



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI PRIRO TOOLS PERHITUNGAN PRIORITISASI PEMBANGUNAN FITUR PADA APLIKASI MENGGUNAKAN METODE PEMBOBOTAN

ANNE ANNISA AULIA
NRP 5114100181

Dosen Pembimbing I
Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc.

Dosen Pembimbing II
Abdul Munif, S.Kom., M.Sc.

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - KI141502

**RANCANG BANGUN APLIKASI PRIRO TOOLS
PERHITUNGAN PRIORITISASI PEMBANGUNAN
FITUR PADA APLIKASI MENGGUNAKAN
METODE PEMBOBOTAN**

**ANNE ANNISA AULIA
NRP 5114100181**

**Dosen Pembimbing I
Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc.**

**Dosen Pembimbing II
Abdul Munif, S.Kom., M.Sc.**

**Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

(Halaman ini sengaja dikosongkan)



UNDERGRADUATE THESES - KI141502

**PRIRO APPLICATION DESIGN AS A TOOL TO
CALCULATE THE PRIORITIZATION ON THE
FEATURE DEVELOPMENT OF AN
APPLICATION USING WIGHTING METHOD**

**ANNE ANNISA AULIA
NRP 5114100181**

First Advisor

Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc.

Second Advisor

Abdul Munif, S.Kom., M.Sc.

**Department of Informatics
Faculty of Information Technology and Communication
Sepuluh Nopember Institute of Technology
Surabaya 2018**

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI PRIRO TOOLS PERHITUNGAN PRIORITISASI PEMBANGUNAN FITUR PADA APLIKASI MENGGUNAKAN METODE PEMBOBOTAN

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

ANNE ANNISA AULIA

NRP : 5114 100 181

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dr.tech.Ir. Raden Venantius Hari Ginandjar,
M.Sc.
NIP. 196505181992031003



2. Abdul Munif, S.Kom., M.Sc.
NIP. 198608232015041004

**SURABAYA
MARET 2018**

(Halaman ini sengaja dikosongkan)

RANCANG BANGUN APLIKASI PRIRO TOOLS PERHITUNGAN PRIORITISASI PEMBANGUNAN FITUR PADA APLIKASI MENGGUNAKAN METODE PEMBOBOTAN

Nama Mahasiswa : Anne Annisa Aulia
NRP : 5114100181
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1: Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.
Dosen Pembimbing 2: Abdul Munif, S.Kom., M.Sc.

Abstrak

Dalam pembangunan sebuah aplikasi ada faktor-faktor yang mampu mempengaruhi jalannya pembangunan aplikasi tersebut. Sering kali perhitungan yang dilakukan di awal sebelum aplikasi dibangun tidak sesuai dengan implementasinya. Tidak sedikit aplikasi yang dibangun dan dirancang dengan apik tetapi aplikasi tersebut bukan merupakan kebutuhan pengguna di lapangan. Fitur-fitur yang dianggap penting bagi pemilik ide atau pembangun aplikasi belum tentu dianggap penting bagi pengguna yang nantinya akan menggunakan aplikasi tersebut.

Aplikasi perhitungan prioritas pembangunan fitur pada aplikasi adalah sebuah aplikasi berbasis web yang dapat membantu pemilik ide atau pembangun aplikasi dalam menentukan fitur-fitur mana yang dianggap penting tidak hanya dari segi developer, namun dari segi pengguna juga. Aplikasi ini akan memberikan hasil berupa urutan fitur yang sudah dikelompokkan berdasarkan tingkat kepentingannya. Perhitungan prioritas ini menggunakan metode pembobotan yaitu Analytical Hierarchy Process dan 100-Dollar-Test.

Perhitungan prioritas akan dipengaruhi oleh beberapa hal yaitu cost berupa waktu dan biaya yang diperlukan dalam pembangunan fitur yang diberikan oleh developer. Persentase

kepentingan fitur bagi desainer pembangunan aplikasi, serta nilai yang diberikan oleh pengguna.

Kata kunci: 100-Dollar-Test, Analytical Hierarchy Process, Aplikasi Berbasis Web, Metode Pembobotan

**PRIRO APPLICATION DESIGN AS A TOOL TO
CALCULATE THE PRIORITIZATION ON THE
FEATURE DEVELOPMENT OF AN APPLICATION
USING WEIGHTING METHOD**

Student's Name : Anne Annisa Aulia
Student's ID : 5114100181
Department : Teknik Informatika FTIF-ITS
First Advisor : Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.
Second Advisor : Abdul Munif, S.Kom., M.Sc.

Abstract

In the development of an application there are factors that affect the development of it. The calculation that happen in the begining of designing the application is not match as the implementation. Not a few application that has been developed and designed nicely but the features of the application is not what user's need. The idea owner might think that the features are important or the designer and developer might think the same way but in reality users may not think the features as important as the developer thinks.

This application for calculating the priority of feature development of an application is a web based application that can help the idea owner or the developer to decide which of the feature is important not only from the developer side but also from the user side. This application will show the result of the features that has been grouped by its imporance. The priority calculation is using weighting methods such as Analytical Hierarchy Process and 100-Dollar-Test..

The priority calculation is affected with some factors which is cost from the time and expense that needed in the development of the feature from the developer. Percentage of feature importance from designer in the development of the application. The last factor is the score that given by the users.

Keywords: 100-Dollar-Test, Analytical Hierarchy Process, Web Based Application, Weighting Method

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

RANCANG BANGUN APLIKASI PRIRO TOOLS PERHITUNGAN PRIORITISASI PEMBANGUNAN FITUR PADA APLIKASI MENGGUNAKAN METODE PEMBOBOTAN

Terselesaikannya Tugas Akhir ini tentunya tidak terlepas dari dukungan, motivasi dan bantuan dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan rasa terima kasih sebesar-besarnya kepada:

1. Tuhan yang Maha Esa atas berkah, rahmat dan hidayah-Nya yang selalu menemani selama pengerjaan Tugas Akhir ini.
2. Ayah dan Ibu penulis, Sugeng Hariyono dan Lusje A. Tabalujan yang tiada hentinya memberikan dukungan penuh baik dari segi doa dan motivasi kepada penulis hingga terselesaikannya Tugas Akhir ini.
3. Kakak Feidy Febryani dan Adik Astrid Nadya Fadilla yang selalu memberikan support dan mendengarkan keluh kesah penulis selama mengerjakan Tugas Akhir ini.
4. Bapak Hari Ginardi dan Bapak Abdul Munif selaku dosen pembimbing yang telah membimbing penulis dengan penuh kesabaran dan memotivasi dalam pengerjaan Tugas Akhir ini.
5. Bapak dan Ibu dosen dan karyawan Jurusan Teknik Informatika ITS yang telah memberikan ilmu dan

pengalaman penulis selama menjadi mahasiswa di Teknik Informatika ITS.

6. Teman-teman ICT Global *Startup* 2017 terutama yang berasal dari Indonesia, Kezia, Mila, Marthin, Sun Win, Tri, Habib, Aldy yang selalu memberikan semangat dan pengertian selama penulis mengerjakan Tugas Akhir ini.
7. Kim Hyung Jun, Wahyu, Luqman, Fatih dan teman-teman lainnya yang menemani penulis mulai dari pengerjaan proposal hingga Tugas Akhir ini.
8. Teman-teman seperjuangan mulai dari pengajuan topik Tugas Akhir ini, Rahmatin Nadia dan Kania Amalia.
9. Anindita, Vinsensia, Avianda dan Tiara yang telah mendengarkan, berbagi, dan menyemangati selama masa perkuliahan penulis.
10. Admin lab Manajemen Informasi, teman-teman TC angkatan 2014 dan kakak-kakak yang selalu memberikan warna dalam keseharian penulis.
11. Serta semua pihak yang tidak bisa disebutkan satu per satu yang selalu membantu, menghibur dan memberikan motivasi baik secara langsung maupun tidak langsung kepada penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa dalam penulisan Tugas Akhir ini masih banyak kekurangannya. Penulis telah berusaha sebaik-baiknya dalam penyusunan Tugas Akhir ini dan penulis memohon maaf atas kekurangan yang terdapat pada Tugas Akhir ini. Penulis mengharapkan kritik dan saran dari pembaca agar menjadi koreksi untuk ke depannya. Terakhir, penulis berharap laporan Tugas Akhir ini berguna bagi pembacanya.

Surabaya,

Anne Annisa Aulia

DAFTAR ISI

Abstrak	vii
Abstract	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Permasalahan.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi.....	3
1.6.1 Penyusunan Proposal Tugas Akhir.....	3
1.6.2 Studi Literatur.....	3
1.6.3 Analisis dan Desain Perangkat Lunak	3
1.6.4 Implementasi Perangkat Lunak	3
1.6.5 Pengujian dan Evaluasi.....	4
1.6.6 Penyusunan Buku	4
1.7 Sistematika Penulisan Laporan.....	4
BAB II TINJAUAN PUSTAKA	7
2.1 Prioritisasi	7
2.2 Metode Prioritisasi Sejenis	8
2.3 <i>Analytic Hierarchy Process (AHP)</i>	9
2.4 <i>100-Dollar-Test</i>	12
2.5 MoSCoW	12
2.6 Kerangka Kerja Laravel.....	13
BAB III ANALISIS DAN PERANCANGAN SISTEM	15
3.1 Analisis Sistem	15
3.1.1 Deskripsi Umum Metode.....	15
3.1.2 Deskripsi Umum Aplikasi	23
3.1.3 Analisis Kebutuhan Sistem.....	24
3.1.4 Analisis Aktor.....	25
3.2 Perancangan Sistem	25
3.2.1 Kasus Penggunaan.....	25
3.2.2 Arsitektur Aplikasi	26

3.2.3	Perancangan Metode terhadap Sistem dan Aplikasi.....	27
3.2.3.1	Rancangan Langkah <i>Product Owner</i> Memberikan Fitur.....	27
3.2.3.1.1	Rancangan Proses Menambah Aplikasi	28
3.2.3.1.2	Rancangan Proses Menambah Fitur	31
3.2.3.2	Rancangan Langkah Desainer Memberikan Persentase Setiap Fitur.....	33
3.2.3.2.1	Rancangan Proses Memberi Persentase Fitur.....	33
3.2.3.3	Rancangan Langkah Pengguna Memberikan Nilai untuk Setiap Fitur.....	37
3.2.3.3.1	Rancangan Proses Menilai Fitur	38
3.2.3.3.2	Rancangan Proses Memberi Waktu dan Biaya Fitur... ..	41
3.2.3.3.3	Rancangan Proses Melihat Hasil Prioritas Pembangunan Aplikasi.....	44
BAB IV IMPLEMENTASI.....		47
4.1	Lingkungan Implementasi	47
4.1.1	Lingkungan Implementasi Perangkat Keras.....	47
4.1.2	Lingkungan Implementasi Perangkat Lunak.....	47
4.2	Implementasi Metode	47
4.2.1.1	Implementasi Pemerolehan Jawaban dari <i>Stakeholder</i>	50
4.2.1.1.1	Implementasi Proses Menambah Aplikasi	50
4.2.1.1.2	Implementasi Proses Menambah Fitur	51
4.2.1.1.3	Implementasi Proses Memberi Persentase Setiap Fitur.	52
4.2.1.1.4	Implementasi Proses Memberi Waktu dan Biaya Fitur.	53
4.2.1.1.5	Implementasi Proses Memberi Nilai Fitur	53
4.2.1.2	Implementasi Perhitungan Nilai <i>Value</i>	54
4.2.1.3	Implementasi Perhitungan Nilai <i>Cost</i>	55
4.2.1.4	Implementasi Pengkategorian Setiap Fitur.....	56
4.3	Implementasi Tampilan Antarmuka	58
4.3.1.1	Implementasi Antarmuka Halaman Awal <i>Product Owner</i>	58
4.3.1.2	Implementasi Antarmuka Halaman Utama <i>Product Owner</i>	58
4.3.1.3	Implementasi Antarmuka Menambah Data Aplikasi	59

4.3.1.4	Implementasi Antarmuka Halaman Detail Aplikasi	60
4.3.1.5	Implementasi Antarmuka Mengubah Data Aplikasi	60
4.3.1.6	Implementasi Antarmuka Menambah Data Fitur ...	61
4.3.1.7	Implementasi Antarmuka Mengubah Data Fitur	62
4.3.1.8	Implementasi Antarmuka Halaman Hasil Prioritas	62
4.3.1.9	Implementasi Antarmuka Detail Hasil Pengguna...	63
4.3.1.10	Implementasi Antarmuka Detail Hasil Desainer	64
4.3.1.11	Implementasi Antarmuka Detail Hasil <i>Developer</i> ..	65
4.3.1.12	Implementasi Antarmuka Halaman Awal Pengguna..	65
4.3.1.13	Implementasi Antarmuka Halaman Menilai.....	66
4.3.1.14	Implementasi Antarmuka Halaman Awal Desainer	66
4.3.1.15	Implementasi Antarmuka Halaman Persentase	67
4.3.1.16	Implementasi Antarmuka Halaman Awal <i>Developer</i> .	68
4.3.1.17	Implementasi Antarmuka Halaman Memberi <i>Cost</i>	68
4.3.1.18	Implementasi Antarmuka Halaman Terima Kasih .	69
4.4	Implementasi Perangkat Lunak	70
4.4.1	Implementasi proses Mengubah Aplikasi.....	70
4.4.2	Implementasi Proses Menghapus Aplikasi	70
4.4.3	Implementasi Proses Mengubah Fitur	71
4.4.4	Implementasi Proses Menghapus Fitur.....	71
4.4.5	Implementasi Proses Perhitungan <i>Value</i> Menggunakan AHP	72
4.4.6	Implementasi Proses Menghitung <i>Cost</i> Menggunakan AHP	74
4.4.7	Implementasi Proses Melihat Hasil Prioritas Aplikasi ...	75
4.4.8	Implementasi Proses Melihat Jawaban Desainer	78
4.4.9	Implementasi proses Melihat Jawaban <i>Developer</i>	78
4.4.10	Implementasi proses Melihat Jawaban Pengguna.....	78
BAB V PENGUJIAN DAN EVALUASI		79
5.1	Lingkungan Pengujian	79
5.2	Skenario Pengujian	79
5.2.1	Pengujian Perancangan Suatu Perangkat Lunak dengan Bantuan Aplikasi	80
5.2.1.1	Pengujian Perancangan Aplikasi CallTenant.....	81
5.2.1.2	Pengujian Perancangan Aplikasi WithYou	84
5.2.2	Pengujian Fungsionalitas	88
5.2.2.1	Pengujian Menambah Aplikasi.....	88

5.2.2.2	Pengujian Mengubah Data Aplikasi.....	89
5.2.2.3	Pengujian Menghapus Aplikasi.....	91
5.2.2.4	Pengujian Menambah Fitur	92
5.2.2.5	Pengujian Mengubah Fitur	93
5.2.2.6	Pengujian Menghapus Fitur	95
5.2.2.7	Pengujian Melihat Jawaban Desainer.....	96
5.2.2.8	Pengujian Melihat Jawaban <i>Developer</i>	97
5.2.2.9	Pengujian Melihat Jawaban Pengguna	98
5.2.2.10	Pengujian Melihat Hasil Prioritas Aplikasi	99
5.2.2.11	Pengujian Memberi Persentase Fitur.....	101
5.2.2.12	Pengujian Memberi Waktu dan Biaya Fitur	102
5.2.2.13	Pengujian Memberi Nilai Fitur	104
5.3	Evaluasi Pengujian	105
5.3.1	Evaluasi Pengujian Perancangan Suatu Perangkat Lunak dengan Bantuan Aplikasi	105
5.3.1.1	Evaluasi Pengujian Perhitungan Metode.....	105
5.3.1.2	Evaluasi Pengujian Perancangan CallTenant	107
5.3.1.3	Evaluasi Pengujian Perancangan WithYou	110
BAB VI KESIMPULAN DAN SARAN		115
6.1	Kesimpulan.....	115
6.2	Saran.....	115
DAFTAR PUSTAKA		117
LAMPIRAN		119
BIODATA PENULIS		121

DAFTAR GAMBAR

Gambar 2.1 Hasil Matriks Normalisasi	11
Gambar 2.2 Langkah Penggunaan <i>100-Dollar-Test</i>	12
Gambar 3.1 Langkah Pengerjaan Prioritisasi	16
Gambar 3.2 Diagram Kasus Penggunaan Aplikasi	26
Gambar 3.3. Arsitektur Sistem Pada Aplikasi.....	27
Gambar 3.4. Diagram Alur <i>Product Owner</i> Memberikan Daftar Fitur .	28
Gambar 3.5 Diagram Aktivitas Menambah Aplikasi	30
Gambar 3.6 Rancangan Antarmuka <i>Form</i> Tambah Aplikasi	30
Gambar 3.7 Diagram Aktivitas Menambah Fitur.....	32
Gambar 3.8 Rancangan Antarmuka Form Tambah Fitur	32
Gambar 3.9. Diagram Alur Desainer Memberikan Persentase Fitur	33
Gambar 3.11 Diagram Aktivitas Memberi Persentase Fitur	36
Gambar 3.12 Rancangan Antarmuka Memberi Persentase Setiap Fitur	37
Gambar 3.13. Diagram Alur Memberikan Nilai Setiap Fitur.....	37
Gambar 3.14 Diagram Aktivitas Menilai Fitur	40
Gambar 3.15 Rancangan Antarmuka Menilai Fitur	41
Gambar 3.16 Diagram Aktivitas Memberi Waktu dan Biaya Fitur	43
Gambar 3.17 Rancangan Antarmuka Memberi Waktu dan Biaya Fitur	44
Gambar 3.18 Diagram Aktivitas Melihat Hasil Prioritas	45
Gambar 3.19 Rancangan Antarmuka Hasil Prioritas	46
Gambar 4.1 Implementasi Perhitungan	48
Gambar 4.2 Implementasi Perhitungan Nilai <i>Value</i>	49
Gambar 4.3 Implementasi Perhitungan Nilai <i>Cost</i>	49
Gambar 4.4 Implementasi Pemerolehan Jawaban dari <i>Stakeholder</i>	50
Gambar 4.5 Halaman Awal <i>Product Owner</i>	58
Gambar 4.6 Halaman Utama <i>Product Owner</i>	59
Gambar 4.7 Halaman Menambah Data Aplikasi.....	60
Gambar 4.8 Halaman Menambah Detail Aplikasi	60
Gambar 4.9 Halaman Mengubah Data Aplikasi	61
Gambar 4.10 Halaman Menambah Data Fitur	61
Gambar 4.11 Halaman Mengubah Data Fitur	62
Gambar 4.12 Halaman Hasil Prioritas.....	63
Gambar 4.13 Halaman Detail Hasil Pengguna.....	64
Gambar 4.14 Halaman Detail Hasil Desainer	64
Gambar 4.15 Halaman Detail Hasil <i>Developer</i>	65
Gambar 4.16 Halaman Awal Pengguna	66
Gambar 4.17 Halaman Menilai	66

Gambar 4.18 Halaman Awal Desainer	67
Gambar 4.19 Halaman Persentase	67
Gambar 4.20 Halaman Awal <i>Developer</i>	68
Gambar 4.21 Halaman Memberi <i>Cost</i>	69
Gambar 4.22 Halaman Terima Kasih	69
Gambar 5.1 Langkah Pengujian Perangkat Lunak / Aplikasi	80
Gambar 5.2 Hasil Prioritas CallTenant	84
Gambar 5.3 Hasil Prioritas WithYou	87
Gambar 5.4 Pengujian Menambah Aplikasi	89
Gambar 5.5 Hasil Pengujian Menambah Aplikasi	89
Gambar 5.6 Pengujian Mengubah Aplikasi	90
Gambar 5.7 Hasil Pengujian Mengubah Aplikasi	91
Gambar 5.8 Pengujian Menghapus Aplikasi	92
Gambar 5.9 Hasil Pengujian Menghapus Aplikasi	92
Gambar 5.10 Pengujian Menambah Fitur	93
Gambar 5.11 Hasil Pengujian Menambah Fitur	93
Gambar 5.12 Pengujian Mengubah Fitur	94
Gambar 5.13 Hasil Pengujian Mengubah Fitur	95
Gambar 5.14 Pengujian Menghapus Fitur	96
Gambar 5.15 Hasil Pengujian Menghapus Fitur	96
Gambar 5.16 Hasil Pengujian Melihat Jawaban Desainer	97
Gambar 5.17 Hasil Pengujian Melihat Jawaban <i>Developer</i>	98
Gambar 5.18 Hasil Pengujian Melihat Jawaban Pengguna	99
Gambar 5.19 Tampilan Awal Melihat Hasil Prioritas Aplikasi	100
Gambar 5.20 Hasil Pengujian Melihat Hasil Prioritas Aplikasi	101
Gambar 5.21 Pengujian Memberi Persentase Fitur	102
Gambar 5.22 Pengujian Memberi Waktu dan Biaya Fitur	103
Gambar 5.23 Pengujian Memberi Nilai Fitur	105
Gambar 5.24 Hasil AHP <i>Value</i> pada Aplikasi	106
Gambar 5.25 Hasil AHP pada Aplikasi Lain	107
Gambar 5.26 Graf Hasil Aplikasi CallTenant	109
Gambar 5.27 Graf Hasil Aplikasi WithYou	112

DAFTAR TABEL

Tabel 2.1 Perbandingan PHandler dan Priro	9
Tabel 2.2 Intensitas Kepentingan	10
Tabel 2.3 Kesesuaian Kategori	13
Tabel 3.1 Kategori <i>Stakeholder</i>	18
Tabel 3.2 Persentase Setiap Fitur oleh Desainer	19
Tabel 3.3 Kepentingan Fitur Bagi Pengguna	20
Tabel 3.4 Waktu dan Biaya Setiap Fitur	20
Tabel 3.5 Matriks <i>Value</i>	21
Tabel 3.6 Matriks <i>Cost</i>	21
Tabel 3.7 Matriks Normalisasi <i>Value</i> dan <i>EigenValue</i> -nya	21
Tabel 3.8 Matriks Normalisasi <i>Cost</i> dan <i>EigenValue</i> -nya	22
Tabel 3.9 Urutan Fitur Sesuai Hasil Perhitungan	22
Tabel 3.10 Daftar Kebutuhan Fungsional Sistem	24
Tabel 3.11 Keterangan Kode Kasus Penggunaan	25
Tabel 3.12 Spesifikasi Kasus Penggunaan Menambah Aplikasi.....	28
Tabel 3.13 Spesifikasi Kasus Penggunaan Menambah Fitur	31
Tabel 3.14 Spesifikasi Kasus Penggunaan Memberi Persentase Fitur ..	34
Tabel 3.15 Spesifikasi Kasus Penggunaan Menilai Fitur.....	38
Tabel 3.16 Spesifikasi Kasus Penggunaan Memberi Waktu dan Biaya Fitur.....	41
Tabel 3.17 Spesifikasi Kasus Penggunaan Melihat Hasil Prioritas Aplikasi.....	44
Tabel 5.1 Demografi Penguji Coba.....	82
Tabel 5.2 Daftar Fitur Aplikasi CallTenant.....	82
Tabel 5.3 Hasil Jawaban Pengguna.....	82
Tabel 5.4 Hasil Jawaban Desainer	83
Tabel 5.5 Hasil Jawaban <i>Developer</i>	83
Tabel 5.6 Demografi Penguji Coba.....	85
Tabel 5.7 Daftar Fitur Aplikasi WithYou	85
Tabel 5.8 Hasil Jawaban Pengguna.....	85
Tabel 5.9 Hasil Jawaban Desainer	86
Tabel 5.10 Hasil Jawaban <i>Developer</i>	86
Tabel 5.11 Pengujian Menambah Aplikasi	88
Tabel 5.12 Pengujian Mengubah Aplikasi	90
Tabel 5.13 Pengujian Menghapus Aplikasi.....	91
Tabel 5.14 Pengujian Menambah Fitur	92
Tabel 5.15 Pengujian Mengubah Fitur	94

Tabel 5.16 Pengujian Menghapus Fitur	95
Tabel 5.17 Pengujian Melihat Jawaban Desainer	96
Tabel 5.18 Pengujian Melihat Jawaban <i>Developer</i>	97
Tabel 5.19 Pengujian Melihat Jawaban Pengguna.....	98
Tabel 5.20 Pengujian Melihat Hasil Prioritas Aplikasi.....	100
Tabel 5.21 Pengujian Memberi Persentase Fitur	101
Tabel 5.22 Pengujian Memberi Waktu dan Biaya Fitur	103
Tabel 5.23 Pengujian Memberi Nilai Fitur	104
Tabel 5.24 Perbandingan Hasil Aplikasi Priro dan Aplikasi Lainnya .	108
Tabel 5.25 Tabel Hasil Aplikasi Priro	108
Tabel 5.26 Hasil Jawaban Desainer	109
Tabel 5.27 Urutan Pembangunan Fitur Berdasarkan Jawaban Desainer Aplikasi CallTenant.....	110
Tabel 5.28 Perbandingan Hasil Aplikasi dan Aplikasi Lainnya	111
Tabel 5.29 Tabel Hasil Aplikasi Priro	112
Tabel 5.30 Hasil Jawaban Desainer	113
Tabel 5.31 Urutan Pembangunan Fitur Berdasarkan Jawaban Desainer Aplikasi WithYou.....	113
Tabel A.1. Jawaban Hasil Kuesioner	119

DAFTAR KODE SUMBER

Kode Sumber 4.1 Fungsi Menambah Aplikasi	51
Kode Sumber 4.2 Fungsi Menambah Fitur	52
Kode Sumber 4.3 Fungsi Memberi Persentase Setiap Fitur	53
Kode Sumber 4.4 Fungsi Memberi Waktu dan Biaya Fitur	53
Kode Sumber 4.5 Fungsi Memberi Nilai Fitur.....	54
Kode Sumber 4.6 Pengubahan Nilai Kepentingan Menjadi Matriks Awal	55
Kode Sumber 4.7 Normalisasi Matriks	55
Kode Sumber 4.8 Perhitungan <i>EigenValue</i>	55
Kode Sumber 4.9 Pengubahan Nilai Kepentingan Menjadi Matriks Awal	56
Kode Sumber 4.10 Normalisasi Matriks	56
Kode Sumber 4.11 Perhitungan <i>EigenValue</i>	56
Kode Sumber 4.12 Pengkategorian Hasil Prioritas	57
Kode Sumber 4.13 Fungsi Mengubah Aplikasi	70
Kode Sumber 4.14 Fungsi Menghapus Aplikasi	71
Kode Sumber 4.15 Fungsi Mengubah Fitur	71
Kode Sumber 4.16 Fungsi Menghapus Fitur	72
Kode Sumber 4.17 Fungsi Perhitungan <i>Value</i> Menggunakan AHP.....	73
Kode Sumber 4.18 Fungsi Menghitung <i>Cost</i> Menggunakan AHP	75
Kode Sumber 4.19 Fungsi Melihat Hasil Prioritas Aplikasi	77

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Siklus pengembangan suatu aplikasi memiliki tahap pengimplementasian aplikasi tersebut. Pengerjaan pembangunan aplikasi tentu tetap dilakukan secara bertahap, mulai dari fitur utama, sekunder hingga fitur tambahan yang dibutuhkan. Sering kali terjadi kekeliruan dalam pembangunan suatu fitur, fitur yang dianggap lebih penting bagi *developer*, belum tentu dianggap sangat penting bagi pengguna. Bahkan, terkadang fitur yang dibangun lebih dahulu tersebut, belum merupakan kebutuhan mendesak yang dibutuhkan oleh pengguna. Memberikan prioritas dalam setiap fitur yang akan dibangun membuat pembangunan aplikasi dilakukan sesuai kebutuhan.

Prioritisasi bisa dilakukan lebih efisien dengan meminta pertimbangan langsung dari *stakeholder* yang akan membangun ataupun mengerjakan aplikasi tersebut. Mulai dari desainer yang merancang fitur-fitur, *developer* yang akan mengerjakan aplikasi tersebut hingga pengguna yang akan menggunakan aplikasinya nanti. Pertimbangan dari setiap *stakeholder* ini akan dijadikan faktor pendukung untuk perhitungan prioritas. Faktor tersebut dibagi menjadi *value* dan *cost*. *Value* di sini adalah nilai kepentingan atau keharusan suatu fitur tersebut dibangun, pertimbangan faktor ini meliputi pertimbangan dari desainer dan calon pengguna aplikasi. Selanjutnya adalah *cost* yang akan didapatkan dari informasi yang diberikan oleh *developer*.

Oleh sebab itu, prioritisasi aplikasi akan dilakukan dengan memperhatikan kedua faktor tersebut. Metode yang digunakan untuk melakukan prioritisasi adalah *Analytic Hierarchy Process* atau AHP.

1.2 Rumusan Masalah

Berdasarkan identifikasi di atas, Tugas Akhir ini mengangkat beberapa rumusan masalah sebagai berikut:

1. Bagaimana nilai kepentingan fitur bagi desainer dapat dipadukan dengan nilai kepentingan bagi pengguna?
2. Bagaimana mendapatkan nilai *cost* dari data yang diperoleh dari *developer*?
3. Bagaimana fitur aplikasi yang akan dikembangkan sesuai kebutuhan pengguna?

1.3 Batasan Permasalahan

Batasan masalah yang terdapat pada Tugas Akhir ini adalah sebagai berikut:

1. Pengambilan kuesioner untuk persentase kepentingan antara desainer dan kepentingan *user* dilakukan secara *online* dan hanya diambil 10 respon tercepat dari pelaku *startup*.
2. Aplikasi Priro hanya menangani perhitungan prioritas untuk pembangunan sebuah aplikasi *startup*.

1.4 Tujuan

Adapun tujuan dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Membuat rancangan agar nilai kepentingan fitur bagi desainer dapat dipadukan dengan nilai kepentingan bagi pengguna
2. Membuat rancangan perhitungan prioritas untuk memperoleh nilai *cost* dari pengolahan data yang diberikan oleh *developer*.
3. Membuat rancangan agar aplikasi dibangun sesuai kebutuhan pengguna dengan memperhatikan faktor dari desainer.

1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah sebagai berikut:

1. Membuat rancangan pembangunan aplikasi yang sesuai dengan kebutuhan pengguna.

2. Membuat rancangan pembangunan aplikasi yang memaksimalkan *value* dan *cost* setiap fiturnya.
3. Membuat rancangan aplikasi yang dapat dikembangkan.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Tahapan awal dari pengerjaan Tugas Akhir ini adalah penyusunan proposal Tugas Akhir. Pada proposal Tugas Akhir ini, penulis mengajukan penggunaan AHP dalam prioritas pembangunan aplikasi.

1.6.2 Studi Literatur

Tahapan selanjutnya adalah tahapan mencari informasi dari studi literatur tentang *Analytical Hierarchy Process* dan *100-Dollar-Test* seperti referensi paper atau referensi lainnya. Di sini referensi yang didapatkan dalam bentuk paper, jurnal, internet, maupun buku materi-materi kuliah.

1.6.3 Analisis dan Desain Perangkat Lunak

Proses analisis berlangsung dari proses *Product Owner* yang akan memasukkan aplikasi yang akan dihitung. Selanjutnya, nilai *value* didapatkan dari pengguna dan desainer, lalu nilai *cost* didapatkan dari nilai yang dimasukkan oleh *developer*.

1.6.4 Implementasi Perangkat Lunak

Pembangunan aplikasi akan dilakukan dengan menggunakan aplikasi berbasis web lokal dengan bahasa pemrograman PHP, HTML, CSS, dan JavaScript. Desain ini juga menggunakan *tools* Sublime Text, *Framework* Laravel, basis data MySQL, dan XAMPP.

1.6.5 Pengujian dan Evaluasi

Pengujian dilakukan oleh *developer* yang akan memasukkan *cost* yang terbagi menjadi lama pengerjaan dan biaya pengerjaan, lalu *Product Owner* memasukkan waktu dan biaya pengerjaan yang akan dilakukan. Pengujian ini dilakukan untuk menguji apakah perhitungan aplikasi sudah sesuai dengan perhitungan prioritas secara manual.

1.6.6 Penyusunan Buku

Pada tahap ini dilakukan penyusunan buku yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibangun.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

Bab I. Pendahuluan

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.

Bab II. Tinjauan Pustaka

Bab ini berisi kajian teori atau penjelasan dari metode, algoritma, *library*, dan *tools* yang digunakan dalam penyusunan Tugas Akhir ini. Bab ini berisi tentang *Analytic Hierarchy Process*, metode *100-Dollar-Test*, bahasa pemrograman PHP, JavaScript dan kerangka kerja Laravel.

Bab III. Perancangan Perangkat Lunak

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan mulai dari perekaman fitur-fitur, kepentingan bagi desainer, kepentingan bagi pengguna, waktu dan biaya dari *developer*, hingga perhitungan AHP dengan *cost* dan *value* pada aplikasi.

Bab IV. Implementasi

Bab ini menjelaskan implementasi dari perancangan dan pengimplementasian setiap fitur pendukung aplikasi.

Bab V. Hasil Uji Coba dan Evaluasi

Bab ini berisikan hasil uji coba fungsionalitas dari fitur-fitur yang telah dibangun serta evaluasi terhadap implementasi yang telah dilakukan.

Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan. Selain itu bab ini juga membahas saran untuk pengembangan aplikasi ke depannya.

Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam Tugas Akhir.

Lampiran

Bab ini merupakan bab tambahan yang berisi daftar pertanyaan beserta rekapannya.

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

Bab ini berisi kajian teori atau penjelasan dari metode, algoritma, *library*, dan *tools* yang digunakan dalam penyusunan Tugas Akhir ini. Bab ini berisi tentang *Analytic Hierarchy Process*, *100-Dollar-Test*, teknik MoSCoW, bahasa pemrograman PHP, JavaScript, dan kerangka kerja Laravel.

2.1 Prioritisasi

Prioritisasi merupakan penyeimbangan keuntungan bisnis dari tiap kebutuhan dengan biaya dan semua implikasi dari kebutuhan tersebut untuk pembangunan pondasi arsitekturalnya maupun evolusi produk di masa mendatang [1]. Dalam pembangunan fitur-fitur pada sebuah aplikasi, prioritisasi pembangunan fitur adalah sebuah proses di mana pembangun aplikasi menemukan kebutuhan-kebutuhan *stakeholder* yang paling penting. Nantinya informasi penting dari setiap fitur ini akan diimplementasikan dalam pembangunan aplikasi tersebut. Selain itu, prioritisasi juga membantu mengidentifikasi faktor yang mampu mempengaruhi dalam perencanaan pembangunan aplikasi ke depannya. Hal ini disebabkan karena, pemilihan kebutuhan yang tepat untuk dibangun juga menjadi pertimbangan yang sangat penting bagi pembangun aplikasi.

Terdapat beberapa teknik prioritisasi yang sudah pernah dilakukan dalam pembangunan aplikasi. Di mana setiap teknik memiliki cara perhitungan dan pendapatan informasi yang berbeda. Beberapa teknik prioritisasi pembangunan aplikasi adalah sebagai berikut:

1. Metode *Ranking*

Metode ini berdasarkan pada pemberian skala yang berurutan kepada daftar yang ingin diberikan prioritas. Misalnya, jika ada n jumlah kebutuhan, maka elemen yang mendapatkan nilai 1 merupakan elemen yang memiliki *ranking* tertinggi atau terpenting. Sementara untuk elemen yang mendapatkan nilai n

atau nilai terakhir dari semua daftar kebutuhan yang ada, maka elemen tersebut merupakan elemen yang paling tidak penting atau memiliki *ranking* terakhir [2]. Namun, metode ini belum bisa mencakup bila ada elemen yang memiliki kepentingan yang setara atau sama pentingnya.

2. Metode *Top-Ten*

Metode ini digunakan untuk membantu menentukan prioritas jika memiliki jumlah elemen yang besar. Pada metode ini setiap *stakeholder* diperkenankan memilih 10 fitur utama dari kebutuhan-kebutuhan yang mereka rasa paling penting. Namun, metode ini mampu menimbulkan konflik di mana hasil dari pemilihan kebutuhan-kebutuhan tersebut jika dibandingkan hasilnya akan bervariasi [2].

2.2 Metode Prioritisasi Sejenis

Priro merupakan aplikasi prioritisasi berbasis web yang mampu memberikan prioritas pembangunan fitur sebuah aplikasi. Paper [1] menjelaskan tentang salah satu cara perhitungan prioritas pembangunan fitur sebuah aplikasi yang telah dilakukan sebelumnya. Pada paper tersebut, penulis menjelaskan mengenai beberapa metode prioritisasi yang sudah ada dan membandingkan beberapa metode. Selain itu, penulis memadukan beberapa metode yang ada menjadi sebuah metode perhitungan prioritas. *PHandler* adalah nama dari metode yang digunakan pada aplikasi tersebut.

Pada metode tersebut, penulis menggunakan metode AHP dan logika Fuzzy. Dari perbandingan beberapa metode yang ada, penulis tetap menggunakan salah satunya yaitu metode AHP, hal ini yang menjadi alasan mengapa dalam pembangunan aplikasi Priro digunakan metode AHP. Aplikasi Priro ini juga ditujukan untuk prioritisasi pembangunan fitur pada aplikasi terutama *startup*. Di mana *startup* hanya memiliki fitur yang sedikit. Seperti halnya proses '*sprint*' yang sering dilakukan dalam inkubator atau pengembangan sebuah *startup*. Pada [3] dijelaskan tentang pembangunan sebuah robot bernama *Savioke* yang ditujukan untuk membantu keperluan hotel. Buku tersebut menjelaskan proses yang

dilakukan untuk mengetahui apakah robot tersebut akan disukai oleh pengunjung hotel atau tidak. Langkah urutan pembangunan robot ini selalu dilakukan secara iterasi hingga menemukan fitur yang diperlukan oleh pengguna.

Dari fakta-fakta di atas, beberapa perbandingan Priro dan aplikasi yang sudah ada dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan PHandler dan Priro

	PHandler	Priro
Metode	AHP dan Fuzzy	AHP, 100-Dollar-Test, Metode MoSCoW
Stakeholder	Tim pembangun aplikasi	Product Owner, Desainer, Developer, Pengguna

2.3 *Analytic Hierarchy Process (AHP)*

Analytic Hierarchy Process atau AHP adalah teknik untuk mendukung proses pengambilan keputusan yang bertujuan untuk menentukan pilihan terbaik dari beberapa alternatif yang dapat diambil[4]. Perhitungan ini mempertimbangkan *cost* dan *value* dari fitur-fitur yang akan dibangun. AHP nantinya diaplikasikan untuk mendapatkan nilai *EigenValue* yang akan digunakan dalam matriks prioritas. AHP akan menghitung persaingan antara setiap kebutuhan pada tiap *cluster* yang ada.

Meskipun ada beberapa metode prioritas yang baru, AHP tetap digunakan karena memiliki akurasi yang lebih baik dalam penentuan prioritas [5]. Perhitungan AHP ini akan digunakan pada perhitungan prioritas tahap akhir untuk mendapatkan prioritas pembangunan aplikasi. Berikut adalah penjelasan mengenai persamaan AHP yang akan digunakan pada pengerjaan Tugas Akhir ini [6]:

- Pemberian *EigenValue* pada setiap fitur terhadap fitur lainnya dengan alternatif perbandingan seperti Tabel 2.2.

Tabel 2.2 Intensitas Kepentingan

Tingkat Kepentingan	Definisi	Penjelasan
1	Setara	Kedua elemen sama pentingnya
3	Sedikit lebih penting	Elemen yang satu sedikit lebih penting dari elemen yang lainnya
5	Lebih penting	Elemen yang satu lebih penting dari elemen yang lainnya
7	Sangat lebih penting	Satu elemen jelas lebih mutlak penting dari pada elemen lainnya
9	Kepentingannya mutlak	Satu elemen mutlak penting dari elemen lainnya
2, 4, 6, 8		Nilai-nilai antara dua pertimbangan yang berdekatan

- Pembentukan matriks dan perhitungannya akan dilakukan seperti Persamaan 2.1:

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \quad (2.1)$$

di mana:

A = Matriks awal

a_{ij} = nilai kepentingan fitur

$$w_i = \sum_{j=1}^n a_{ij} = [w_1 \quad \dots \quad w_n] \quad (2.2)$$

di mana:

n = jumlah fitur

w = jumlah nilai perfitur

$$\text{MatriksNormal} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \cdot [1/w_1 \quad \dots \quad 1/w_n] = \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{nn} \end{bmatrix} \quad (2.3)$$

di mana:

MatriksNormal = Hasil matriks normalisasi

b_{ij} = nilai kepentingan per fitur yang sudah dinormalisasi

$$EigenValue = \frac{1}{n} \cdot \sum_{j=1}^n b_{ij} = [e_1 \quad \dots \quad e_n] \quad (2.4)$$

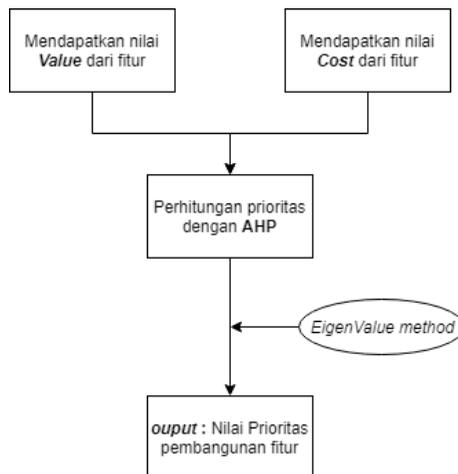
di mana:

EigenValue = Nilai *EigenValue* per fitur e_i = *EigenValue* per fitur

Persamaan 2.1, 2.2, 2.3, dan 2.4 adalah langkah mendapatkan nilai *EigenValue* menggunakan AHP yang akan dilakukan dengan:

- (2.1) Menentukan intensitas kepentingan setiap fitur terhadap fitur lainnya.
- (2.2) Intensitas tersebut akan dijadikan sebuah matriks.
- (2.3) Matriks tersebut akan dinormalisasikan.
- (2.4) Matriks terbaru akan dinormalisasikan kembali menggunakan prinsip *EigenValue*.

Persamaan tersebut akan diaplikasikan pada tahapan perhitungan seperti pada Gambar 2.1. Di mana dalam pengaplikasiannya perhitungan AHP dilakukan secara dinamis, atau bergantung pada hasil jawaban *stakeholder*.

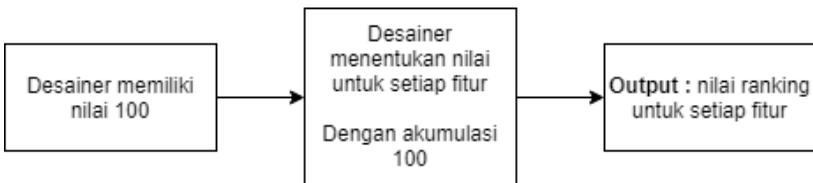


Gambar 2.1 Hasil Matriks Normalisasi

2.4 100-Dollar-Test

100-Dollar-Test adalah teknik memberikan *ranking* untuk setiap fitur. Metode ini dimulai dengan seolah-olah memberikan 100 dolar kepada desainer, dan meminta mereka untuk mengisikan ke mana uang tersebut akan mereka habiskan dalam setiap fitur yang ada [2].

Perhitungan *100-Dollar-Test* ini akan diimplementasikan dalam menentukan prioritas dari desainer.



Gambar 2.2 Langkah Penggunaan 100-Dollar-Test

Penjelasan dari Gambar 2.2 adalah:

1. Desainer memiliki nilai 100 yang dapat digunakan untuk memberi nilai setiap fitur.
2. Desainer memberikan nilai pada setiap fitur dengan akumulasi 100.
3. Seluruh hasil pemberian *ranking* yang dilakukan desainer akan dirata-rata untuk mendapatkan urutan *ranking* fitur bagi desainer.

2.5 MoSCoW

Teknik MoSCoW adalah salah satu teknik penentuan prioritas yang berasal dari *Dynamic Software Development Method* (DSDM) [7]. Teknik MoSCoW ini merupakan salah satu cara untuk mengetahui pendapat dari *stakeholder*. Cara kerja dari teknik ini adalah dengan memberikan pendapat untuk setiap elemen. Pendapat yang akan diberikan pada setiap elemen dibagi menjadi empat kategori yaitu:

1. *Must Have*, didefinisikan sebagai elemen yang direkomendasikan harus dimiliki atau merupakan elemen yang sangat penting.
2. *Should*, merupakan bagian dari elemen yang direkomendasikan untuk dimiliki. Elemen ini bukan termasuk yang sangat vital atau sangat penting, tetapi akan lebih baik jika memiliki elemen ini.
3. *Could*, pada bagian ini elemen merupakan kebutuhan yang *nice-to-have* atau baik untuk dimiliki tetapi tidak terlalu diharapkan. Bukan merupakan elemen penting dari kebutuhan yang ada.
4. *Won't* atau *Would*, elemen ini adalah elemen yang paling rendah tingkat diharapkannya. Jika dibandingkan dengan ketiga kategori lainnya, kategori ini menjadi bagian yang paling sedikit prioritasnya karena elemen ini hanya berpengaruh sedikit terhadap kebutuhan atau bahkan tidak sama sekali.

Dalam pengerjaan Tugas Akhir ini teknik MoSCoW digunakan untuk meminta pendapat dari pengguna untuk setiap fiturnya. Namun, penggunaan kategori dari teknik ini akan disesuaikan dengan kebutuhan. Pertanyaan yang akan ditanyakan dan kesesuaian kategori dari teknik MoSCoW dapat dilihat pada Tabel 2.3.

Tabel 2.3 Kesesuaian Kategori

Kategori	Pernyataan
<i>Must Have</i>	Sangat Dibutuhkan
<i>Should</i>	Dibutuhkan
<i>Could</i>	Mungkin Dibutuhkan
<i>Won't / Would</i>	Tidak Dibutuhkan

2.6 Kerangka Kerja Laravel

Laravel adalah suatu kerangka kerja dalam pengembangan *website* yang menggunakan konsep MVC (*model-view-controller*). Laravel dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya

pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu [8].

Pada pengerjaan Tugas Akhir ini, kerangka kerja Laravel digunakan untuk visualisasi kepada calon pengguna aplikasi. Mulai dari perekaman fitur, perekaman jawaban dari *stakeholder* hingga hasil prioritas akan ditunjukkan dalam bentuk web dan menggunakan kerangka kerja Laravel. Konsep dari arsitektur Laravel sendiri, MVC dapat dijelaskan lebih lanjut sebagai berikut:

- a. *Model*, digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada perubahan informasi [9]. *Model* digunakan sebagai tempat yang akan mengakses *database*.
- b. *View*, bertanggung jawab untuk pemetaan grafis ke sebuah perangkat [9]. *View* biasanya memiliki hubungan 1-1 dengan sebuah permukaan layar dan pada aplikasi Priro *view* ini adalah bagian yang berhubungan langsung dengan pengguna.
- c. *Controller*, menerima input dari pengguna dan menginstruksikan *model* dan *view* untuk melakukan aksi berdasarkan masukan tersebut. Sehingga, *controller* bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi [9]. Priro menggunakan *controller* juga sebagai penghubung antara *model* dan *view*.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tentang perancangan perangkat lunak. Selain itu, bab ini juga menjelaskan tentang analisis, perancangan, dan pembuatan sistem yang akan dibangun. Perancangan sistem yang dibangun mulai dari perekaman fitur-fitur, kepentingan bagi desainer, kepentingan bagi pengguna, waktu dan biaya dari *developer*, hingga perhitungan AHP dengan *cost* dan *value* pada aplikasi. Selain itu, perancangan sistem yang akan dibangun akan direpresentasikan menggunakan diagram.

3.1 Analisis Sistem

Pada tahap analisis sistem ini yang akan dibahas meliputi analisis masalah, analisis kebutuhan, deskripsi umum sistem, dan kasus penggunaan sistem yang dibangun.

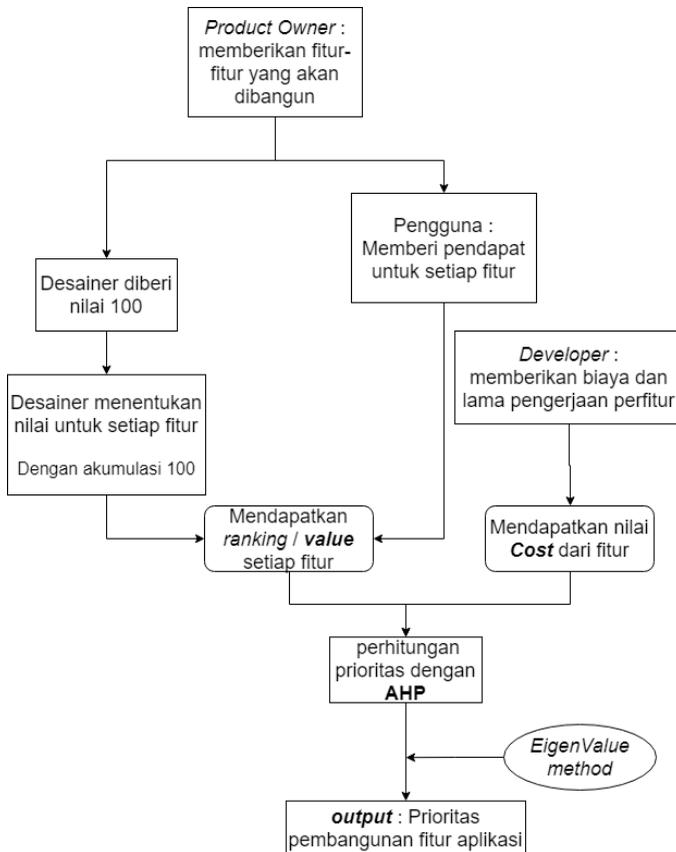
3.1.1 Deskripsi Umum Metode

Aplikasi ini akan menghitung prioritas pembangunan menggunakan *Analytic Hierarchy Process* dan *100-Dollar-Test*. Kasus yang dapat ditangani oleh Tugas Akhir ini contohnya pembangunan sebuah aplikasi *startup*. Di mana pada awal pembangunan aplikasi tersebut akan dimulai dari fitur-fitur penting dan yang dibutuhkan pengguna, dilanjutkan dengan beberapa fitur pendukung. Agar pembangunan aplikasi tersebut sesuai sasaran, dibutuhkan prioritas untuk setiap fiturnya.

Prioritisasi ini dibutuhkan karena dalam beberapa pengerjaan suatu aplikasi biasanya terdapat kasus di mana fitur yang paling dibutuhkan oleh pengguna justru dibangun paling terakhir. Bahkan dalam beberapa kasus pembangunan suatu aplikasi, fitur pendukung justru menjadi daya tarik terbesar bagi para pengguna. Sehingga, untuk kasus-kasus khusus seperti itu dibutuhkan prioritas agar pembangunannya tidak akan memakan banyak biaya dan yang dikerjakannya pun sesuai target pengguna. Tugas Akhir ini akan membantu menyelesaikan masalah pembangunan aplikasi tersebut

dengan prioritas menggunakan AHP dan *100-Dollar-Test*. Penggunaan aplikasi Priro ini dapat dilakukan mulai dari tahap awal atau analisis dan perancangan pembangunan aplikasi.

Langkah-langkah yang diajukan dalam penyelesaian masalah aplikasi dapat dilihat pada Gambar 3.1.



Gambar 3.1 Langkah Pengerjaan Prioritisasi

Penjelasan dari Gambar 3.1 adalah sebagai berikut:

1. Merancang kuesioner untuk pelaku *startup* yang akan digunakan untuk persentase perhitungan *value*

berdasarkan kepentingan dari desainer dan kepentingan dari pengguna.

2. Melakukan perekaman fitur-fitur yang akan dibangun dalam aplikasi oleh *Product Owner*. Selain itu kita harus mendapatkan nilai *value* dari desainer. Dengan metode *ranking* yaitu *100-Dollar-Test*, di mana setiap fitur akan diurutkan berdasarkan kepentingannya bagi *stakeholder*. Penjelasan tahapan metode *ranking* ini adalah:
 - i. Desainer memiliki nilai 100 yang dapat digunakan untuk memberi nilai setiap fitur.
 - ii. Desainer memberikan nilai pada setiap fitur dengan akumulasi 100.
 - iii. Seluruh hasil pemberian *ranking* yang dilakukan desainer akan dirata-rata untuk mendapatkan urutan *ranking* fitur bagi *stakeholder*.
3. Mendapatkan nilai kepentingan dari pengguna dengan menanyakan seberapa setuju mereka dengan fitur yang akan dibangun menggunakan metode MoSCoW. Setiap pengguna akan mendapatkan 4 pilihan untuk setiap fitur, yaitu:
 - i. Sangat dibutuhkan
 - ii. Dibutuhkan
 - iii. Mungkin dibutuhkan
 - iv. Tidak dibutuhkan

Setelah kuesioner disebar, jumlah penjawab setiap pilihan akan dijumlahkan. Nilai *ranking* dari desainer akan dihitung dengan hasil kuesioner yang didapat dari pengguna, untuk mendapatkan nilai *ranking* akhir yang akan menjadi *value* untuk prioritisasi.

4. Nilai *cost* didapatkan dari *developer* yang menginputkan biaya dan waktu yang diperlukan untuk membangun setiap fitur.
5. Sistem akan menghitung prioritas untuk setiap fiturnya menggunakan AHP.

6. Sistem menampilkan hasil prioritas pembangunan aplikasi *startup*.

Stakeholder yang mampu memberikan pendapat mereka untuk setiap fiturnya dibagi menjadi tiga kategori yang dapat dilihat pada Tabel 3.1. Selain itu, aktor lainnya yang akan menggunakan aplikasi Priro adalah *Product Owner*, di mana *Product Owner* merupakan orang yang memberikan penjelasan mengenai aplikasi serta fiturnya yang akan dihitung prioritas pembangunannya.

Tabel 3.1 Kategori Stakeholder

No	Nama	Karakteristik
1.	Desainer	<ul style="list-style-type: none">• Orang yang merancang pembangunan dari aplikasi, biasanya dibutuhkan pada tahapan analisis dan desain.• Mampu memberikan persentase kepentingan kepada setiap fitur.
2.	<i>Developer</i>	<ul style="list-style-type: none">• Orang yang melakukan pembangunan aplikasi• Memberikan biaya dan waktu pengerjaan untuk setiap fitur.
3.	Pengguna	<ul style="list-style-type: none">• Klien dari aplikasi atau orang yang akan menggunakan aplikasi.• Memberikan nilai atau pendapat tentang kepentingan setiap fitur.

Hasil akhir dari Tugas Akhir ini berupa aplikasi Priro yang akan merekam fitur-fitur dari *startup* yang akan dibangun. Untuk sebuah gambaran pengerjaan Tugas Akhir ini, berikut ini merupakan contoh pembangunan aplikasi pengirim pesan.

Langkah 1. Pada langkah ini kuesioner diberikan kepada pelaku *startup* untuk mendapatkan pembobotan *value* dari desainer dan pengguna. Sebab dalam perancangan sistem ini pengambilan kuesioner belum dilaksanakan, kita dapat memisalkan kepentingan dari desainer diberi bobot 50% dan pengguna 50%. Angka ini

nantinya akan digunakan untuk pendapatan nilai kepentingan *value* total.

Langkah 2. Untuk aplikasi pengiriman pesan ini dibutuhkan beberapa fitur, yaitu:

1. Mengirim dan menerima pesan
2. Memberikan pilihan emoji
3. Dapat mengirimkan gambar
4. Dapat mengirimkan suara
5. Dapat menyembunyikan pesan

Dari kelima fitur tersebut desainer memberikan *ranking* kepentingan untuk setiap fiturnya menggunakan *100-Dollar-Test*. Data yang didapat ditunjukkan pada Tabel 3.2.

Tabel 3.2 Persentase Setiap Fitur oleh Desainer

Fitur	Desainer 1	Desainer 2	Desainer 3	Desainer 4	Desainer 5	Total / 500%
1	50%	60%	30%	75%	70%	0.57
2	15%	10%	20%	5%	10%	0.12
3	15%	10%	20%	10%	10%	0.13
4	10%	10%	20%	5%	10%	0.11
5	10%	10%	10%	5%	0%	0.07
Total	100%	100%	100%	100%	100%	1

Langkah 3. Untuk mendapatkan nilai *value* dibutuhkan beberapa pengguna yang akan memberikan pendapatnya untuk setiap fitur. Pada langkah ini teknik MoSCoW akan digunakan. Penjelasan untuk keempat kategori adalah SD untuk Sangat Dibutuhkan, D untuk Dibutuhkan, MD untuk Mungkin Dibutuhkan dan TD untuk Tidak Dibutuhkan. Pada langkah ini dapat dimisalkan pendapat diperoleh dari 5 pengguna aplikasi pengiriman pesan.

Tabel 3.3 Kepentingan Fitur Bagi Pengguna

Fitur	SD	D	MD	TD
1	0	0	0	5
2	0	0	2	3
3	0	0	3	2
4	0	0	5	0
5	0	4	1	0

Berdasarkan Tabel 3.3 didapatkan *ranking* fitur berdasarkan keinginan pengguna. Dari langkah 2 dan 3, nilai *ranking* per fitur akan dihitung untuk mendapatkan *value* dengan persentase yang didapatkan dari langkah 1. Misalkan setelah dihitung fitur 1 mendapatkan nilai 8, fitur 2 dan 3 mendapatkan nilai 6, fitur 4 mendapatkan 5, dan fitur ke 5 mendapatkan nilai 0.

Langkah 4. Perhitungan *cost* didapatkan dari *developer* yang memberikan biaya serta waktu yang dibutuhkan dalam mengerjakan setiap fiturnya. Misalkan *developer* memberikan biaya dan waktu seperti Tabel 3.4.

Tabel 3.4 Waktu dan Biaya Setiap Fitur

Fitur	Biaya	Waktu (hari)
1	10.000.000	10
2	2.000.000	8
3	5.000.000	2
4	4.000.000	6
5	7.000.000	4

Setelah dihitung, nilai dari *cost* per hari berdasarkan Tabel 3.4, adalah untuk fitur 1 mendapatkan nilai 1.000.000 per hari, fitur 2 mendapatkan nilai 250.000.000 per hari, fitur 3 mendapatkan nilai 2.500.000 per hari, fitur 4 mendapatkan nilai 666.666 perhari, dan fitur ke 5 mendapatkan nilai 1.750.000 per hari.

Langkah 5. Untuk menghitung prioritas menggunakan AHP kita membutuhkan nilai *value* dan *cost*. Berdasarkan langkah ke 3 dan ke 4, kita dapat mengubahnya menjadi matriks seperti Tabel 3.5 dan Tabel 3.6.

Tabel 3.5 Matriks Value

Fitur	1	2	3	4	5
1	1	3	3	4	9
2	0.333	1	1	2	7
3	0.333	1	1	2	7
4	0.25	0.5	0.5	1	6
5	0.111	0.143	0.143	0.167	1

Tabel 3.6 Matriks Cost

Fitur	1	2	3	4	5
1	1	4	0.2	2	0.333
2	0.25	1	0.125	0.333	0.167
3	5	8	1	6	3
4	0.5	3	0.167	1	0.25
5	3	6	0.333	4	1

Setelah memiliki matriks awal untuk *cost* dan *value*, kita harus melakukan normalisasi dan perhitungan *EigenValue* pada setiap fitur dari tabelnya. Dengan menggunakan contoh Tabel 3.5 dan Tabel 3.6, kita akan mendapatkan tabel normalisasi matriks *value*, *cost* dengan nilai *EigenValue* masing-masing yang dapat dilihat pada Tabel 3.7 dan Tabel 3.8.

Tabel 3.7 Matriks Normalisasi Value dan EigenValue-nya

Fitur	1	2	3	4	5	<i>EigenValue</i>
1	0.4932	0.5316	0.5316	0.4364	0.3	0.458561092
2	0.1644	0.1772	0.1772	0.2182	0.2333	0.194065819
3	0.1644	0.1772	0.1772	0.2182	0.2333	0.194065819
4	0.1233	0.0886	0.0886	0.1091	0.2	0.121918754
5	0.0548	0.0253	0.0253	0.0182	0.0333	0.031388517

Tabel 3.8 Matriks Normalisasi Cost dan EigenValue-nya

Fitur	1	2	3	4	5	<i>EigenValue</i>
1	0.1026	0.1818	0.1096	0.15	0.0702	0.122829353
2	0.0256	0.0455	0.0685	0.025	0.0351	0.039935288
3	0.5128	0.3636	0.5479	0.45	0.6316	0.501196206
4	0.0513	0.1364	0.0913	0.075	0.0526	0.081320294
5	0.3077	0.2727	0.1826	0.3	0.2105	0.25471886

Langkah 6. Setelah mendapatkan *EigenValue* kita bisa membagi tingkat prioritas menjadi 4 kategori yaitu:

1. Fitur dengan *value* tinggi dan *cost* rendah
2. Fitur dengan *value* tinggi dan *cost* tinggi
3. Fitur dengan *value* rendah dan *cost* rendah
4. Fitur dengan *value* rendah dan *cost* tinggi

Tingkat prioritas dibagi menjadi 4 kategori seperti di atas dengan penjelasan, fitur-fitur yang memiliki kategori pertama atau fitur dengan *value* yang tinggi dan *cost* yang rendah akan menjadi prioritas yang utama. Hal ini dikarenakan fitur tersebut memiliki nilai kepentingan yang tinggi atau sangat dibutuhkan dan *cost* pengerjaannya pun rendah. Sementara itu, kategori yang keempat merupakan fitur yang paling rendah tingkat kepentingannya untuk dibangun karena pada kategori ini fitur juga memiliki *cost* yang tinggi.

Hasil prioritas akan ditampilkan kepada *user* dalam bentuk urutan daftar fitur sesuai prioritas seperti pada Tabel 3.9.

Tabel 3.9 Urutan Fitur Sesuai Hasil Perhitungan

Urutan	Nama Fitur	% <i>Value</i>	% <i>Cost</i>
1.	Mengirim dan menerima pesan	45.86%	12.29%
2.	Memberikan pilihan emoji	19.41%	3.99%
3.	Dapat mengirimkan suara	19.41%	50.12%
4.	Dapat menyembunyikan pesan	12.20%	8.13%
5.	Dapat mengirimkan gambar	3.14%	25.47%

3.1.2 Deskripsi Umum Aplikasi

Aplikasi Priro yang dibangun ini merupakan aplikasi berbasis web yang dibuat untuk membantu *Product Owner* atau pembangun aplikasi untuk menentukan urutan fitur pada aplikasi yang akan dibangun. Penentuan urutan pembangunan fitur pada aplikasi ini juga dipengaruhi oleh pendapat dari beberapa pihak yaitu desainer, *developer*, dan pengguna.

Dalam pengumpulan data yang akan digunakan untuk perhitungan prioritas, setiap pihak diberikan hak akses yang berbeda dan kebutuhan informasi yang berbeda. Pada aplikasi ini *Product Owner* akan berperan sebagai aktor yang memberikan informasi terkait aplikasi yang akan dihitung prioritas pembangunan untuk setiap fiturnya. Selanjutnya, informasi yang diberikan akan dijadikan acuan untuk menjadi pertanyaan kepada aktor lainnya.

Aktor desainer diberikan akses untuk memberikan persentase pada setiap fitur yang akan dibangun. Total akumulasi dari persentasenya adalah 100% dan harus dibagi kepada seluruh fitur. Persentase dari desainer ini menjadi salah satu faktor untuk menghitung nilai *value* pada metode pembobotan. Faktor kedua dalam perhitungan *value* adalah pendapat dari pengguna. Aktor pengguna akan memberikan pendapatnya pada setiap fitur. Tingkat kepentingan setiap fitur harus diberikan pendapat mengenai seberapa pentingkah fitur tersebut untuk dibangun.

Aktor terakhir adalah *developer*, di mana *developer* yang memberikan nilai *cost* yang digunakan dalam perhitungan prioritas. *Developer* memberikan data mengenai biaya dan lama waktu pengerjaan fitur.

Selanjutnya, setelah mendapatkan nilai kepentingan untuk *cost* dan *value* akan dilakukan perhitungan AHP yang nantinya akan menjadi nilai *EigenValue* untuk setiap fiturnya. Nilai *EigenValue* ini yang akan menentukan prioritas pembangunan untuk setiap fitur aplikasi tersebut. Kategorinya akan dibagi dari *value* tinggi dengan *cost* rendah, *value* tinggi dengan *cost* tinggi, *value* rendah dengan *cost* rendah hingga *value* rendah dengan *cost* tinggi.

3.1.3 Analisis Kebutuhan Sistem

Kebutuhan utama dari sistem ini adalah sistem mampu menghitung dan memberikan hasil berupa prioritas pembangunan sebuah aplikasi dengan urutan fitur-fiturnya. Analisis kebutuhan fungsional dari aplikasi Priro dapat dilihat pada Tabel 3.10.

Tabel 3.10 Daftar Kebutuhan Fungsional Sistem

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-001	Menambah Aplikasi	<i>Product Owner</i> mampu menambah aplikasi yang ingin dihitung prioritasnya
F-002	Mengubah Aplikasi	<i>Product Owner</i> mampu mengubah data aplikasi yang sudah terekam pada <i>database</i>
F-003	Menghapus Aplikasi	<i>Product Owner</i> mampu menghapus data aplikasi yang sudah terekam pada <i>database</i>
F-004	Menambah Fitur	<i>Product Owner</i> mampu menambah fitur untuk setiap aplikasi
F-005	Mengubah Fitur	<i>Product Owner</i> mampu mengubah informasi fitur yang sudah terekam pada <i>database</i>
F-006	Menghapus Fitur	<i>Product Owner</i> mampu menghapus fitur yang sudah terekam pada <i>database</i>
F-007	Melihat Jawaban Desainer	<i>Product Owner</i> mampu melihat hasil jawaban dari desainer
F-008	Melihat Jawaban Developer	<i>Product Owner</i> mampu melihat hasil jawaban dari <i>developer</i>
F-009	Melihat Jawaban Pengguna	<i>Product Owner</i> mampu melihat hasil jawaban dari pengguna
F-010	Melihat Hasil Prioritas Aplikasi	<i>Product Owner</i> mampu melihat hasil perhitungan prioritas pembangunan aplikasi
F-011	Memberi Persentase Fitur	Desainer mampu memberikan persentase untuk setiap fitur
F-012	Memberi Waktu	<i>Developer</i> mampu memberikan data

	dan Biaya Fitur	berupa biaya dan lama pengerjaan setiap fitur
F-013	Memberi Nilai Fitur	Pengguna dapat memberikan nilai atau pendapat kepada setiap fitur

3.1.4 Analisis Aktor

Aktor adalah pihak-pihak yang akan berinteraksi secara langsung dengan sistem. Aplikasi Priro memiliki empat aktor yaitu *Product Owner*, *Desainer*, *Developer*, dan *Pengguna*.

3.2 Perancangan Sistem

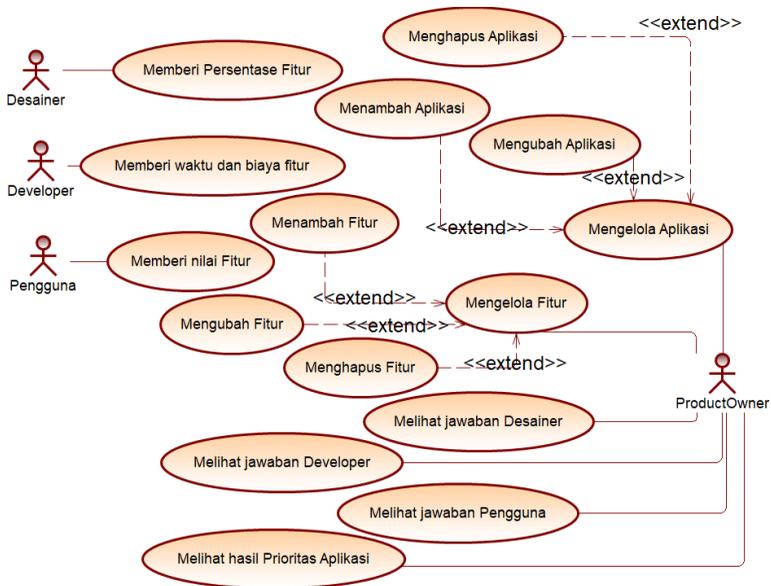
Tahap perancangan sistem dibagi menjadi beberapa bagian, yaitu kasus penggunaan, perancangan arsitektur aplikasi, dan perancangan antarmuka aplikasi.

3.2.1 Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisis kebutuhan fungsional yang telah dilakukan sebelumnya pada Tabel 3.11, sedangkan diagram kasus penggunaan dapat dilihat pada Gambar 3.2.

Tabel 3.11 Keterangan Kode Kasus Penggunaan

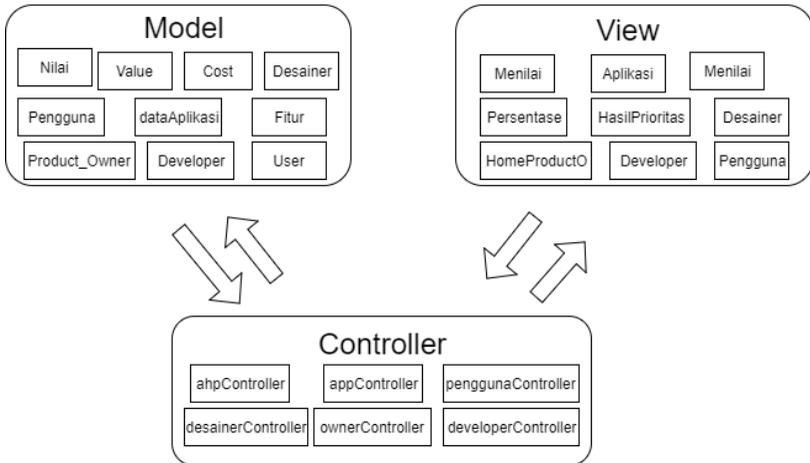
Kode Kasus Penggunaan	Kebutuhan Fungsional
UC-001	Menambah Aplikasi
UC-002	Mengubah Aplikasi
UC-003	Menghapus Aplikasi
UC-004	Menambah Fitur
UC-005	Mengubah Fitur
UC-006	Menghapus Fitur
UC-007	Melihat Jawaban <i>Desainer</i>
UC-008	Melihat Jawaban <i>Developer</i>
UC-009	Melihat Jawaban <i>Pengguna</i>
UC-010	Melihat Hasil Prioritas Aplikasi
UC-011	Memberi Persentase Fitur
UC-012	Memberi Waktu dan Biaya Fitur
UC-013	Memberi Nilai Fitur



Gambar 3.2 Diagram Kasus Penggunaan Aplikasi

3.2.2 Arsitektur Aplikasi

Aplikasi prioritas ini adalah aplikasi berbasis web dengan *Database Server* untuk menyimpan data. Penggunaan *framework* Laravel dengan konsep MVC menjadikan aplikasi ini memiliki arsitektur yang dijelaskan pada Gambar 3.3.



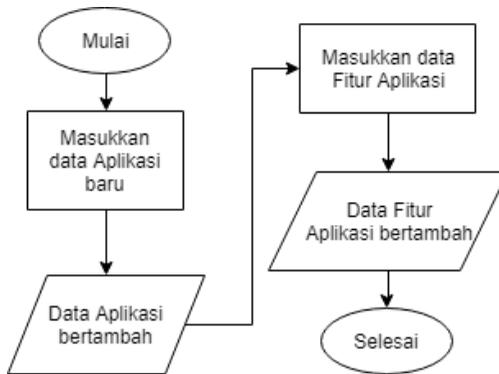
Gambar 3.3. Arsitektur Sistem Pada Aplikasi

3.2.3 Perancangan Metode terhadap Sistem dan Aplikasi

Subbab ini menjelaskan bagaimana metode akan bekerja pada sistem dan aplikasi. Pada subbab ini metode yang sudah dirancang di awal Bab 3 akan diberikan rancangan sistemnya sesuai kasus penggunaan pada aplikasi. Selain itu, rancangan antarmuka untuk jalannya metode juga dijelaskan pada subbab ini.

3.2.3.1 Rancangan Langkah *Product Owner* Memberikan Fitur

Pada langkah ini *Product Owner* dapat memberikan daftar fitur dari aplikasi yang ingin dihitung nilai prioritas fiturnya. Fitur ini lah yang nantinya menjadi dasar pertanyaan untuk pengguna, desainer, dan *developer*. Rancangan langkah *Product Owner* untuk memberikan fitur pada aplikasi dapat dilihat pada Gambar 3.4. Sebelum diberikan daftar fitur yang akan dikerjakan, penambahan aplikasi harus dilakukan karena dalam perhitungannya akan dilakukan per aplikasi dengan setiap daftar fitur di dalamnya. Penjelasan rancangan langkah ini pada aplikasi dijelaskan pada subbab di bawah.



Gambar 3.4. Diagram Alur *Product Owner* Memberikan Daftar Fitur

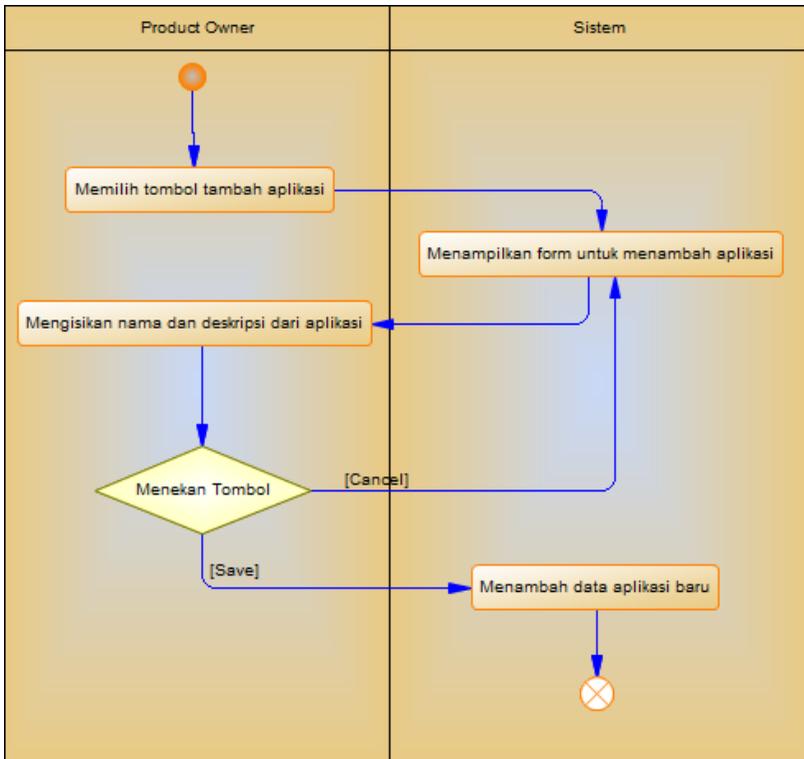
3.2.3.1.1 Rancangan Proses Menambah Aplikasi

Pada proses ini, *Product Owner* mampu melakukan penambahan data aplikasi yang baru. Spesifikasi kasus penggunaan dari proses ini dapat dilihat pada Tabel 3.12 dan diagram aktivitas pada Gambar 3.5. Perancangan untuk antarmuka proses ini dapat dilihat pada Gambar 3.6. Antarmuka akan mencakup tempat untuk mengisikan nama aplikasi baru, penjelasan aplikasi, dan tempat untuk memberikan fitur-fitur aplikasi beserta penjelasannya.

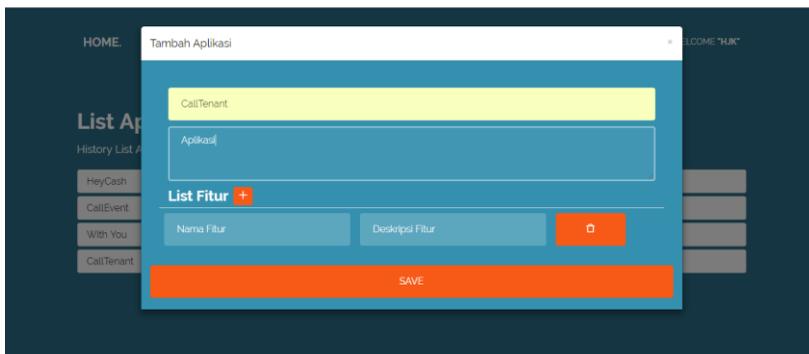
Tabel 3.12 Spesifikasi Kasus Penggunaan Menambah Aplikasi

Kode	UC-001	
Nama	Menambah Aplikasi	
Deskripsi	<i>Product Owner</i> mampu menambah aplikasi yang ingin dihitung prioritasnya	
Tipe	Fungsional	
Pemicu	<i>Product Owner</i> memilih tombol tambah aplikasi	
Aktor	<i>Product Owner</i>	
Kondisi Awal	<i>Product Owner</i> telah melakukan <i>login</i>	
Kondisi Akhir	Data aplikasi baru berhasil terekam ke dalam <i>database</i>	
Alur Kejadian Normal	<i>Product Owner</i>	Sistem
	1. Memilih tombol tambah aplikasi	

		2. Menampilkan <i>form</i> untuk menambah aplikasi
	3. Mengisikan nama dan deskripsi dari aplikasi 4. Menekan tombol <i>Save</i> A.4. <i>Product Owner</i> menekan tombol 'x' (<i>cancel</i>)	
		5. Menambah data aplikasi baru
Alur Kejadian Alternatif	A.4. <i>Product Owner</i> menekan tombol 'x' (<i>cancel</i>) A.4.1. Kembali ke alur kejadian nomor 2	



Gambar 3.5 Diagram Aktivitas Menambah Aplikasi



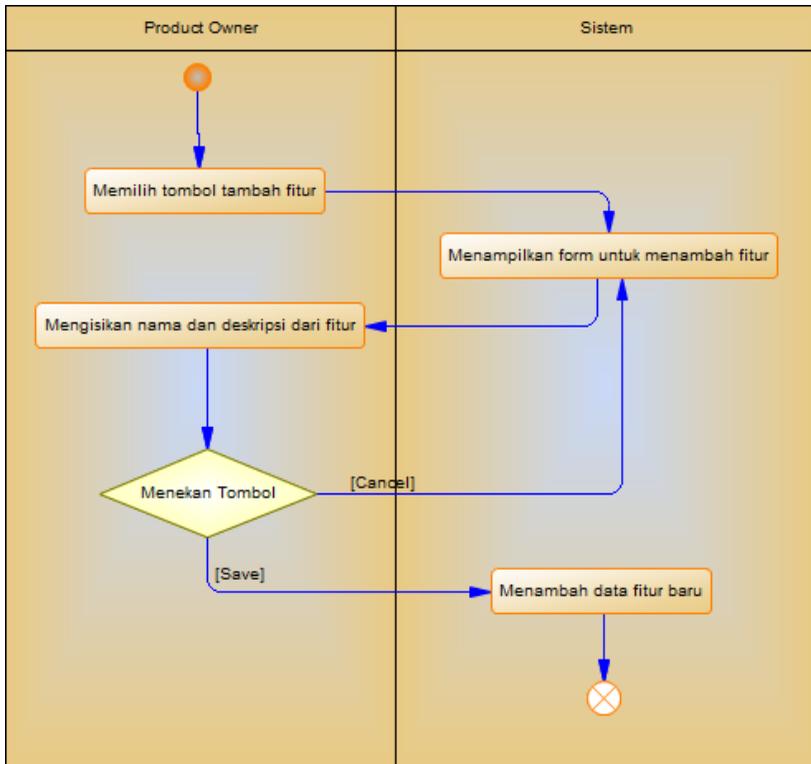
Gambar 3.6 Rancangan Antarmuka *Form* Tambah Aplikasi

3.2.3.1.2 Rancangan Proses Menambah Fitur

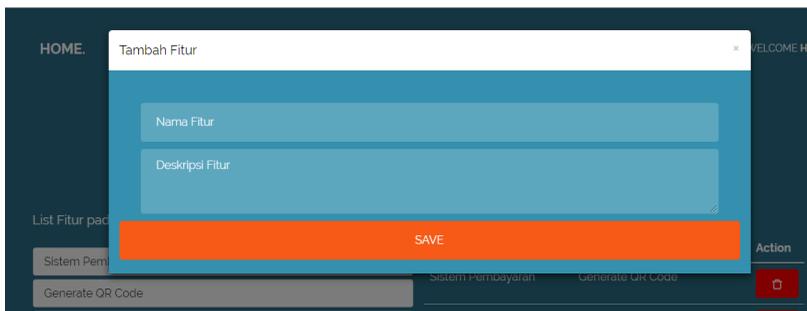
Pada proses ini, *Product Owner* mampu melakukan penambahan data fitur yang baru. Spesifikasi kasus penggunaan dari proses ini dapat dilihat pada Tabel 3.13 dan diagram aktivitas pada Gambar 3.7. Perancangan untuk antarmuka proses ini dapat dilihat pada Gambar 3.8. Antarmuka akan mencakup tempat untuk mengisikan nama fitur baru dan penjelasan fitur.

Tabel 3.13 Spesifikasi Kasus Penggunaan Menambah Fitur

Kode	UC-004	
Nama	Menambah Fitur	
Deskripsi	<i>Product Owner</i> mampu menambah fitur untuk setiap aplikasi	
Tipe	Fungsional	
Pemicu	<i>Product Owner</i> memilih tombol tambah fitur	
Aktor	<i>Product Owner</i>	
Kondisi Awal	<i>Product Owner</i> berada pada halaman aplikasi	
Kondisi Akhir	Data fitur yang baru terekam pada <i>database</i>	
Alur Kejadian Normal	<i>Product Owner</i>	Sistem
	1. Memilih tombol tambah fitur	
		2. Menampilkan <i>form</i> untuk menambah fitur
	3. Mengisikan nama dan deskripsi dari fitur 4. Menekan tombol <i>Save</i> A.4. Menekan tombol 'x' (<i>cancel</i>)	
		5. Menambah data fitur baru
Alur Kejadian Alternatif	A.4. <i>Product Owner</i> menekan tombol 'x' (<i>cancel</i>) A.4.1. Kembali ke alur kejadian nomor 2	



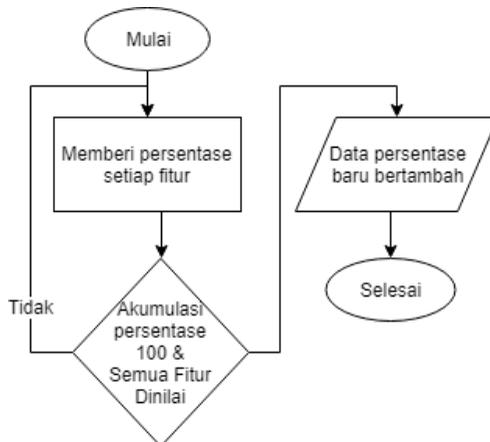
Gambar 3.7 Diagram Aktivitas Menambah Fitur



Gambar 3.8 Rancangan Antarmuka *Form* Tambah Fitur

3.2.3.2 Rancangan Langkah Desainer Memberikan Persentase Setiap Fitur

Pada langkah ini desainer dapat memberikan persentase kepentingan untuk setiap fitur. Nilai ini yang nantinya digunakan untuk menghitung nilai *value* dari setiap fitur. Rancangan langkah desainer untuk memberikan persentase pada setiap fitur dapat dilihat pada Gambar 3.9. Dalam pemberian persentase, setiap fitur harus dipastikan data yang dimasukkan benar dan akumulasi dari persentase semua fitur dalam suatu aplikasi adalah 100%. Metode *100-Dollar-Test* digunakan pada langkah ini, rumus dari metode dapat dilihat pada Persamaan 3.1.



Gambar 3.9. Diagram Alur Desainer Memberikan Persentase Fitur

$$\sum_{i=1}^n \text{PersentaseFitur}_i = 100 \quad (3.1)$$

di mana:

i = indeks fitur

n = jumlah fitur

3.2.3.2.1 Rancangan Proses Memberi Persentase Fitur

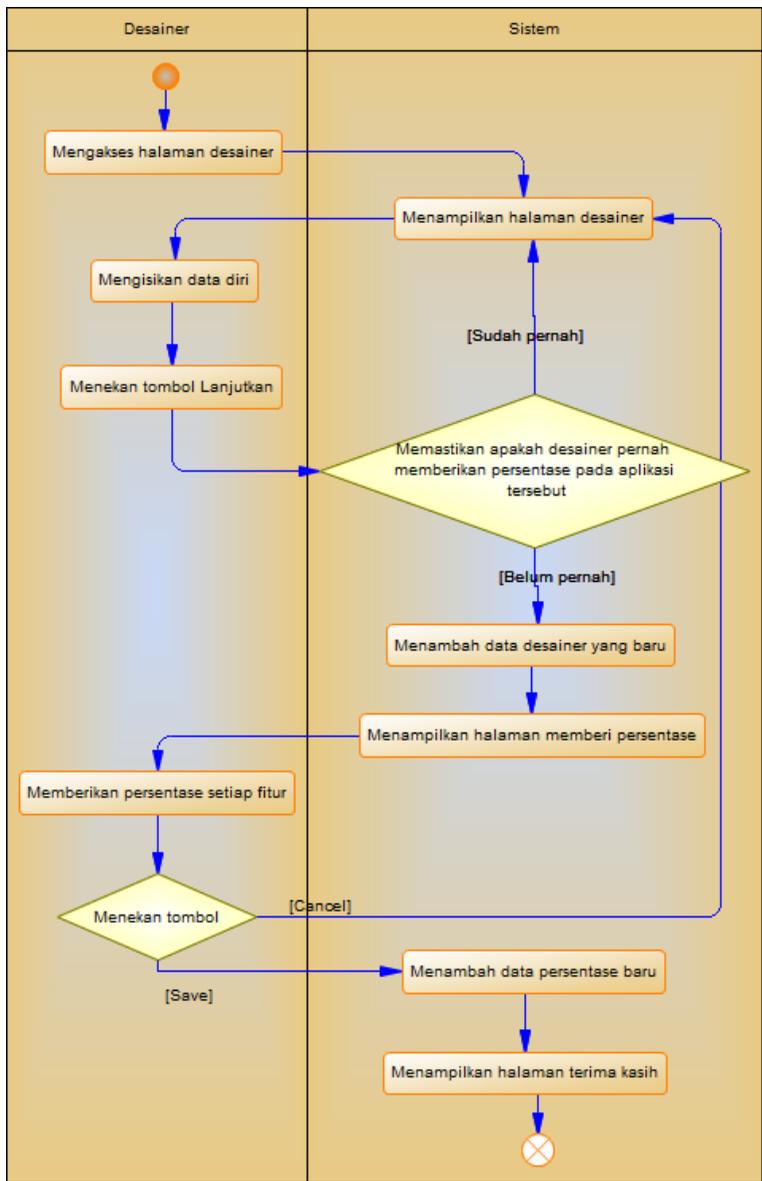
Pada proses ini, desainer mampu memberikan persentase pada setiap fitur. Spesifikasi kasus penggunaan dari proses ini dapat dilihat pada Tabel 3.14 dan diagram aktivitas pada Gambar 3.10.

Perancangan untuk antarmuka proses ini dapat dilihat pada Gambar 3.11.

Tabel 3.14 Spesifikasi Kasus Penggunaan Memberi Persentase Fitur

Kode	UC-011	
Nama	Memberi Persentase Fitur	
Deskripsi	Desainer mampu memberikan persentase untuk setiap fitur	
Tipe	Fungsional	
Pemicu	Desainer mengisi data diri pada halaman desainer	
Aktor	Desainer	
Kondisi Awal	Desainer mengakses halaman untuk desainer	
Kondisi Akhir	Sistem menyimpan data persentase yang diberikan oleh desainer ke dalam <i>database</i>	
Alur Kejadian Normal	Desainer	Sistem
	1. Mengakses halaman desainer	2. Menampilkan halaman desainer
	3. Mengisikan data diri 4. Menekan tombol Lanjutkan	
		5. Memastikan desainer belum pernah memberi persentase pada aplikasi tersebut A.5. Mendapati Desainer pernah memberikan persentase pada aplikasi yang sama 6. Menambah data desainer yang baru 7. Menampilkan halaman memberi persentase
	8. Memberikan persentase pada setiap fitur	

	9. Menekan tombol <i>Save</i> A.8. Menekan tombol <i>Cancel</i>	
		10. Menambah data persentase yang baru ke dalam <i>database</i> 11. Menampilkan halaman terima kasih
Alur Kejadian Alternatif	A.5. Sistem mendapati Desainer pernah memberikan persentase pada aplikasi yang sama A.5.1. Kembali ke alur kejadian nomor 2 A.8. Desainer menekan tombol <i>Cancel</i> A.8.1. Kembali ke alur kejadian nomor 2	



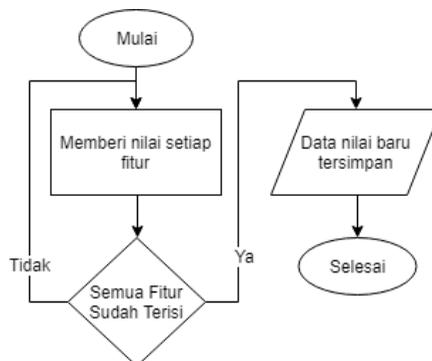
Gambar 3.10 Diagram Aktivitas Memberi Persentase Fitur

- CallEvent -		
Aplikasi yang mempertemukan antara Event dan Tenant		
Hi Hapus Anda akan memberikan persentase kepentingan pada setiap fitur CallEvent		
Total akumulasi persentase adalah 100!		
Nama Fitur	Deskripsi Fitur	Nilai
Mencari Event	pada fitur ini user mampu mencari dan melihat informasi mengenai event yang ada	ex:8%
Mencari Tenant	EO dapat mencari tenant yang diinginkan untuk diajak bergabung pada event	ex:8%
Sistem Pembayaran	Aplikasi mampu menangani sistem pembayaran dari tenant ke EO	ex:8%

Gambar 3.11 Rancangan Antarmuka Memberi Persentase Setiap Fitur

3.2.3.3 Rancangan Langkah Pengguna Memberikan Nilai untuk Setiap Fitur

Pada langkah ini Pengguna dapat memberikan nilai untuk setiap fitur. Nilai ini yang nantinya digunakan untuk menghitung nilai *value* dari setiap fitur. Rancangan langkah pengguna untuk memberikan nilai pada setiap fitur dapat dilihat pada Gambar 3.12. Dalam pemberian nilai harus dipastikan semua fitur telah diberikan nilai. Teknik MoSCoW digunakan pada langkah ini.



Gambar 3.12. Diagram Alur Memberikan Nilai Setiap Fitur

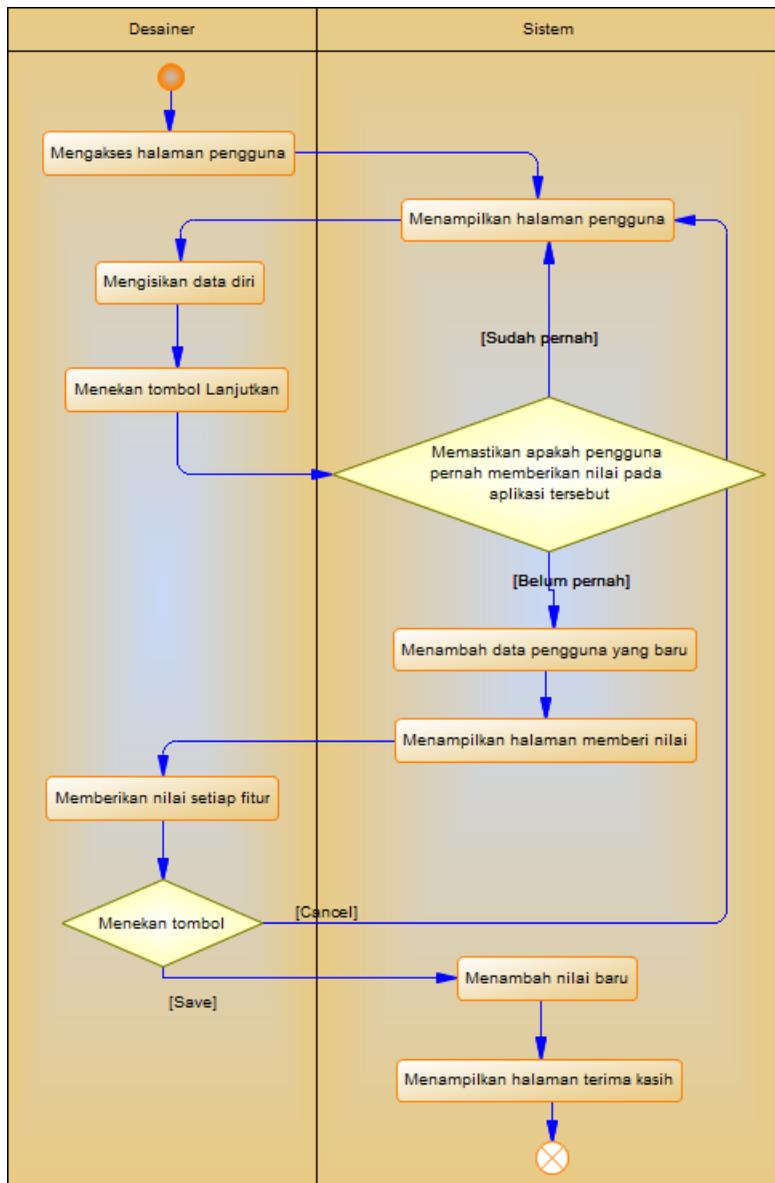
3.2.3.3.1 Rancangan Proses Menilai Fitur

Pada proses ini, pengguna mampu memberikan nilai pada setiap fitur. Spesifikasi kasus penggunaan dari proses ini dapat dilihat pada Tabel 3.15 dan diagram aktivitas pada Gambar 3.13. Perancangan untuk antarmuka proses ini dapat dilihat pada Gambar 3.14. Antarmuka akan mencakup nama fitur dan penjelasannya, lalu tombol untuk pengguna memilih pendapatnya tentang fitur tersebut.

Tabel 3.15 Spesifikasi Kasus Penggunaan Menilai Fitur

Kode	UC-013	
Nama	Memberi Nilai Fitur	
Deskripsi	Pengguna dapat memberikan nilai atau pendapat kepada setiap fitur	
Tipe	Fungsional	
Pemicu	Pengguna mengisi data diri pada halaman pengguna	
Aktor	Pengguna	
Kondisi Awal	Pengguna mengakses halaman untuk pengguna	
Kondisi Akhir	Sistem menyimpan nilai yang diberikan oleh pengguna ke dalam <i>database</i>	
Alur Kejadian Normal	Pengguna	Sistem
	1. Mengakses halaman pengguna	
		2. Menampilkan halaman pengguna
	3. Mengisikan data diri 4. Menekan tombol Lanjutkan	
		5. Memastikan pengguna belum pernah memberi nilai pada aplikasi tersebut A.5. Mendapati pengguna pernah memberikan nilai pada aplikasi yang sama 6. Menambah data pengguna yang baru ke

		dalam <i>database</i> 7. Menampilkan halaman memberi nilai
	8. Memberikan nilai pada setiap fitur 9. Menekan tombol <i>Save</i> A.8. Menekan tombol <i>Cancel</i>	
		10. Menambah nilai yang baru ke dalam <i>database</i> 11. Menampilkan halaman terima kasih
Alur Kejadian Alternatif	A.5. Sistem mendapati pengguna pernah memberikan nilai pada aplikasi yang sama A.5.1. Kembali ke alur kejadian nomor 2 A.8. Pengguna menekan tombol <i>Cancel</i> A.8.1. Kembali ke alur kejadian nomor 2	



Gambar 3.13 Diagram Aktivitas Menilai Fitur



Gambar 3.14 Rancangan Antarmuka Menilai Fitur

3.2.3.3.2 Rancangan Proses Memberi Waktu dan Biaya Fitur

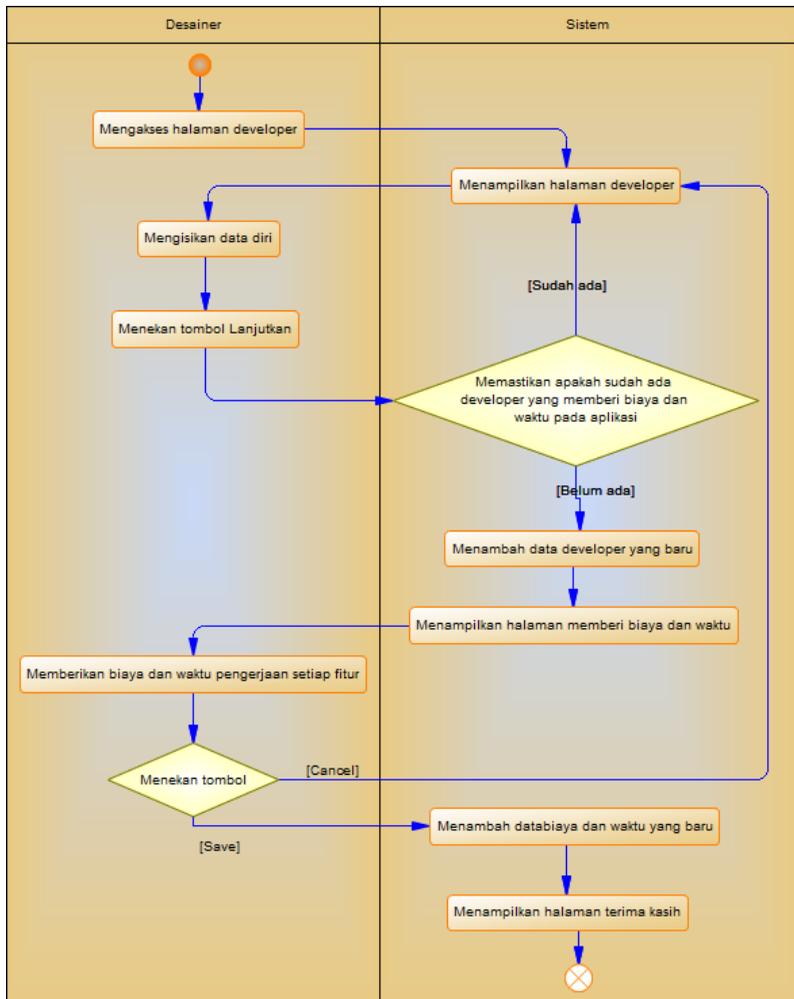
Pada proses ini, *developer* mampu memberikan waktu dan biaya pengerjaan pada setiap fitur. Waktu dan biaya yang diberikan ini digunakan untuk perhitungan *cost* pada setiap fitur. Spesifikasi kasus penggunaan dari proses ini dapat dilihat pada Tabel 3.16 dan diagram aktivitas dapat dilihat pada Gambar 3.15. Perancangan untuk antarmuka proses ini dapat dilihat pada Gambar 3.16. Antarmuka akan mencakup nama fitur dan penjelasannya, lalu tempat untuk *developer* memberikan waktu dan biaya pengerjaan setiap fitur.

Tabel 3.16 Spesifikasi Kasus Penggunaan Memberi Waktu dan Biaya Fitur

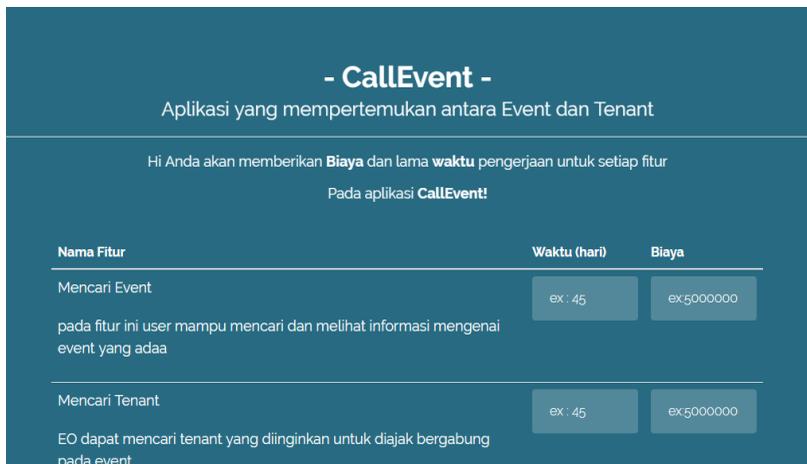
Kode	UC-012
Nama	Memberi Waktu dan Biaya Fitur
Deskripsi	<i>Developer</i> mampu memberikan data berupa biaya dan lama pengerjaan setiap fitur
Tipe	Fungsional
Pemicu	<i>Developer</i> mengisi data diri pada halaman <i>developer</i>
Aktor	<i>Developer</i>
Kondisi Awal	<i>Developer</i> mengakses halaman untuk <i>developer</i>

Kondisi Akhir	Sistem menyimpan data persentase yang diberikan oleh <i>developer</i> ke dalam <i>database</i>	
Alur Kejadian Normal	<i>Developer</i>	Sistem
	1. Mengakses halaman <i>developer</i>	
		2. Menampilkan halaman <i>developer</i>
	3. Mengisikan data diri 4. Menekan tombol Lanjutkan	
		5. Memastikan belum pernah ada <i>developer</i> yang memberi biaya dan waktu pengerjaan pada aplikasi tersebut A.5. Sistem mendapati sudah adanya biaya dan waktu pengerjaan pada aplikasi yang sama 6. Menambah data <i>developer</i> yang baru ke dalam <i>database</i> 7. Menampilkan halaman memberi biaya dan waktu
	8. Memberikan biaya dan waktu pengerjaan pada setiap fitur 9. Menekan tombol <i>Save</i> A.8. <i>Developer</i> menekan tombol <i>Cancel</i>	
		10. Menambah data biaya dan waktu pengerjaan yang baru ke dalam <i>database</i> 11. Menampilkan halaman terima kasih
Alur	A.5 Sistem mendapati sudah adanya biaya dan waktu	

Kejadian Alternatif	<p>pengerjaan pada aplikasi yang sama</p> <p>A.5.1. Kembali ke alur kejadian nomor 2</p> <p>A.8. <i>Developer</i> menekan tombol <i>Cancel</i></p> <p>A.8.1. Kembali ke alur kejadian nomor 2</p>
---------------------	---



Gambar 3.15 Diagram Aktivitas Memberi Waktu dan Biaya Fitur



Gambar 3.16 Rancangan Antarmuka Memberi Waktu dan Biaya Fitur

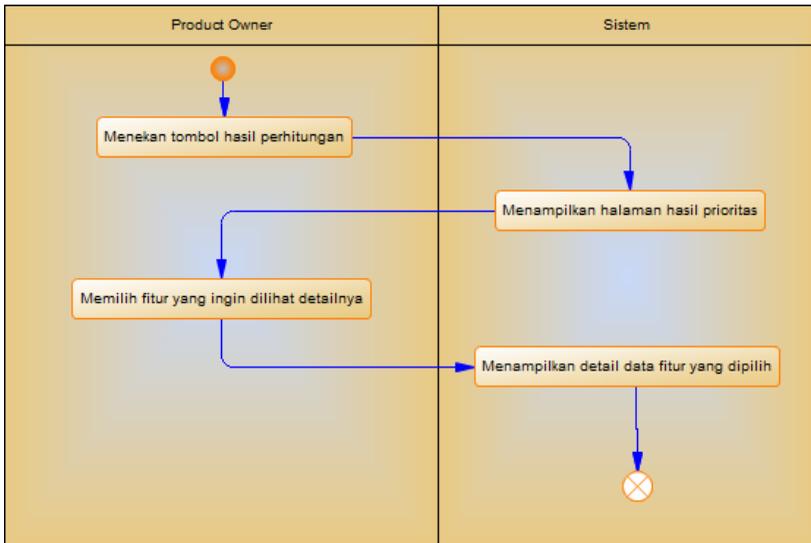
3.2.3.3.3 Rancangan Proses Melihat Hasil Prioritas Pembangunan Aplikasi

Pada proses ini, *Product Owner* mampu melihat hasil urutan fitur yang telah dihitung prioritasnya. Selain melihat jawaban hasil prioritas aplikasi, detail jawaban dari desainer, *developer*, dan pengguna juga dapat dilihat pada proses ini. Spesifikasi kasus penggunaan dari proses ini dapat dilihat pada Tabel 3.17 dan diagram aktivitas pada Gambar 3.17. Perancangan antarmuka proses ini dapat dilihat pada Gambar 3.18. Antarmuka akan menampilkan urutan prioritas untuk fitur-fitur dari aplikasi.

Tabel 3.17 Spesifikasi Kasus Penggunaan Melihat Hasil Prioritas Aplikasi

Kode	UC-010
Nama	Melihat Hasil Prioritas Aplikasi
Deskripsi	<i>Product Owner</i> mampu melihat hasil perhitungan prioritas pembangunan aplikasi
Tipe	Fungsional
Pemicu	<i>Product Owner</i> memilih tombol hasil prioritas
Aktor	<i>Product Owner</i>

Kondisi Awal	<i>Product Owner</i> berada pada halaman aplikasi	
Kondisi Akhir	Sistem menampilkan urutan fitur setelah dilakukan perhitungan	
Alur Kejadian Normal	<i>Product Owner</i>	Sistem
	1. Menekan tombol hasil perhitungan	
		2. Menampilkan halaman hasil prioritas
	3. Memilih fitur yang ingin dilihat detailnya	
		4. Menampilkan detail data fitur yang dipilih
Alur Kejadian Alternatif		



Gambar 3.17 Diagram Aktivitas Melihat Hasil Prioritas

CallTenant

HASIL PENGGUNA

HASIL PROGRAMMER

HASIL DESAINER

Urutan prioritas pembangunan aplikasi *CallTenant*

Fitur dengan *Value Tinggi* dan *Cost Rendah*

1. Mencari Tenant

Fitur dengan *Value Tinggi* dan *Cost Tinggi*

2. Mencari Event

Fitur dengan *Value Rendah* dan *Cost Rendah*

3. Manage Tenant

4. Registrasi Booth

5. Review

Fitur dengan *Value Rendah* dan *Cost Tinggi*

6. Sistem Pembayaran

Gambar 3.18 Rancangan Antarmuka Hasil Prioritas

BAB IV IMPLEMENTASI

Bab ini akan membahas implementasi sistem sesuai dengan analisis dan perancangan sistem yang sudah dilakukan pada bab sebelumnya. Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, kode sumber utama, dan hasil implementasi antarmuka.

4.1 Lingkungan Implementasi

Dalam pengimplementasiannya, aplikasi ini membutuhkan beberapa perangkat pendukung baik dari segi perangkat keras dan perangkat lunak yang akan dijelaskan pada subbab berikut.

4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan aplikasi ini adalah sebagai berikut:

Jenis	: Laptop
Tipe	: Asus A455L
Prosesor	: Intel® Core™ i7-4510U (3.60 GHz)
Memori/RAM	: 4 GB

4.1.2 Lingkungan Implementasi Perangkat Lunak

Perangkat Lunak yang digunakan dalam implementasi aplikasi ini adalah sebagai berikut:

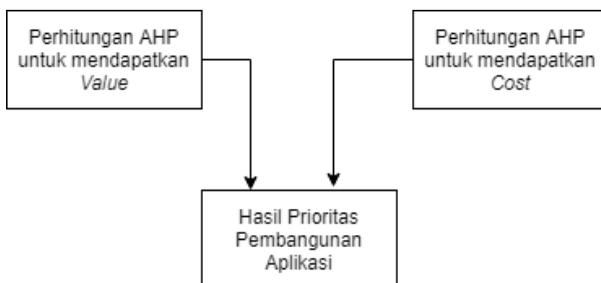
1. Microsoft Windows 10 Pro sebagai sistem operasi.
2. MySQL untuk mengimplementasikan rancangan basis data.
3. SublimeText sebagai *text editor*.
4. Apache sebagai *web server*.
5. PowerDesigner untuk merancang diagram-diagram.

4.2 Implementasi Metode

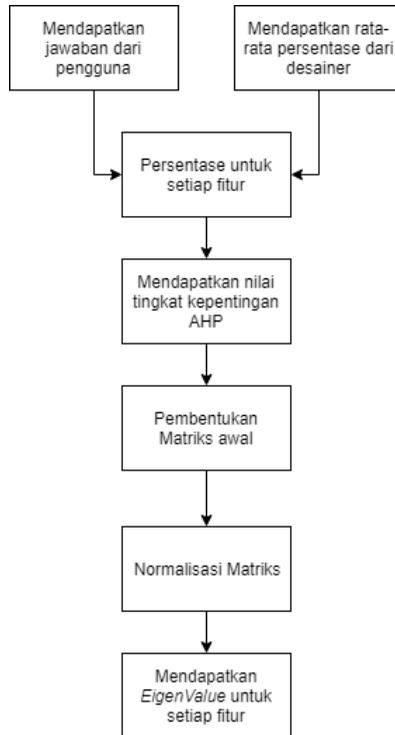
Subbab ini menjelaskan tentang pengimplementasian metode yang telah dirancang pada Bab 3. Metode *Analytical Hierarchy Process* dan *100-Dollar-Test* merupakan dua metode

utama yang digunakan dalam pengerjaan aplikasi ini. Implementasi metode dilakukan seperti langkah perhitungan prioritas yang dijelaskan pada Gambar 3.1.

Dalam pengimplementasiannya perhitungan dilakukan secara bertahap, tahapan yang dimaksud dapat dilihat pada Gambar 4.1, di mana gambar tersebut menjelaskan tentang pengimplementasian tahapan perhitungan. Dengan detail penjelasan urutan pengerjaan pada Gambar 4.2 yang menjelaskan tentang urutan perhitungan *value* dengan lebih rinci dan Gambar 4.3 yang menjelaskan tentang urutan perhitungan *cost* dengan lebih rinci.



Gambar 4.1 Implementasi Perhitungan



Gambar 4.2 Implementasi Perhitungan nilai *Value*



Gambar 4.3 Implementasi Perhitungan nilai *Cost*

Karena perhitungan *value* dipengaruhi dua faktor yaitu jawaban dari pengguna dan desainer, maka kuesioner dibutuhkan untuk mengetahui persentase kepentingan bagi pengguna dan desainer. Hasil kuesioner diperoleh dari 10 responden dan didapati bahwa persentase untuk pengguna dan desainer masing-masing adalah 62,5% dan 37,5%. Angka ini digunakan untuk mendapatkan

persentase keseluruhan untuk setiap fitur. Lalu dari persentase tersebut, didapatkan nilai tingkat kepentingan AHP yang kemudian dijadikan matriks awal.

4.2.1.1 Implementasi Pemerolehan Jawaban dari *Stakeholder*

Pemerolehan jawaban dari *stakeholder* adalah langkah awal yang harus dilakukan sebelum perhitungan prioritas karena untuk menghitung nilai *value* ataupun nilai *cost* dibutuhkan jawaban dari pengguna, *developer*, dan desainer. Setidaknya, setiap elemen telah memiliki satu data jawaban untuk digunakan dalam perhitungan.

Namun, sebelum mendapatkan jawaban dari *stakeholder* tentunya dibutuhkan data mengenai aplikasi dan fitur yang akan diberikan nilai, persentase ataupun waktu dan biaya pengerjaannya. Urutan implementasi dari proses ini dapat dilihat pada Gambar 4.4.



Gambar 4.4 Implementasi Pemerolehan Jawaban dari *Stakeholder*

4.2.1.1.1 Implementasi Proses Menambah Aplikasi

Pengimplementasian dari proses menambah aplikasi bertujuan untuk perekaman data aplikasi baru yang akan dihitung

prioritas pembangunan aplikasinya. Implementasi proses ini menggunakan fungsi **newApp()** yang akan menambahkan data aplikasi yang baru ke dalam *database*. Untuk menjalankan fungsi ini, *Product Owner* harus menekan tombol + yaitu tombol untuk tambah aplikasi, lalu mengisikan data aplikasi tersebut. Setelah tombol *Save* ditekan, sistem akan menuju fungsi **newApp()** yang berada pada *appController*. Fungsi **newApp()** ini akan menambahkan data aplikasi yang baru seperti yang dapat dilihat pada Kode Sumber 4.1.

1.	<code>public function newApp(Request \$request){</code>
2.	<code> \$app = new Aplikasi;</code>
3.	<code> \$app->id_owner = \$request->idid;</code>
4.	<code> \$app->nama_aplikasi = \$request->namaApp;</code>
5.	<code> \$app->deskripsi_aplikasi = \$request->desApp;</code>
6.	<code> \$app->save();</code>
7.	
8.	<code> \$id=Aplikasi::orderBy('id_aplikasi','desc')->first();</code>
9.	<code> \$idApp=\$id->id_aplikasi;</code>
10.	
11.	<code> foreach(\$request->namaFit as \$index => \$nama) {</code>
12.	<code> \$fitur= new Fitur;</code>
13.	<code> \$fitur->id_aplikasi=\$idApp;</code>
14.	<code> \$fitur->nama_fitur=\$nama;</code>
15.	<code> \$fitur->deskripsi_fitur=\$request->desFit[\$index];</code>
16.	<code> \$fitur->save();</code>
17.	<code> }</code>
18.	<code> return Redirect('/app');</code>
19.	<code>}</code>

Kode Sumber 4.1 Fungsi Menambah Aplikasi

4.2.1.1.2 Implementasi Proses Menambah Fitur

Proses ini dilakukan oleh *Product Owner* untuk menambahkan data fitur. Implementasi proses ini adalah menggunakan fungsi **newFitur()** yang akan menambahkan data fitur yang baru ke dalam *database*. Untuk menjalankan fungsi ini, *Product Owner* harus menekan tombol tambah fitur, lalu mengisikan data fitur tersebut. Setelah tombol *Save* ditekan, sistem akan menuju fungsi **newFitur()** yang berada pada *appController*. Fungsi

newFitur() ini akan menambahkan data fitur yang baru seperti yang dapat dilihat pada Kode Sumber 4.2.

```
1. public function newFitur(Request $request)
2. {
3.     $fitur= new Fitur;
4.     $fitur->id_aplikasi=$request->idApp;
5.     $fitur->nama_fitur=$request->namaFitur;
6.     $fitur->deskripsi_fitur=$request->desFitur;
7.     $fitur->save();
8.     return back();
9. }
```

Kode Sumber 4.2 Fungsi Menambah Fitur

4.2.1.1.3 Implementasi Proses Memberi Persentase Setiap Fitur

Proses ini akan menangani perekaman persentase yang diberikan oleh desainer. Fungsi yang digunakan untuk proses ini adalah **addMemberivalue()** pada *desainerController*. Fungsi ini akan merekam persentase jika jumlahnya sudah mencapai 100. Jika tidak maka fungsi ini akan kembali kepada halaman pemberian persentase setiap fitur. Kode program fungsi **addMemberivalue()** dapat dilihat pada Kode Sumber 4.3.

```
1. public function addMemberivalue(Request $request){
2.     $i=0;
3.     foreach ($request->persenFitur as $key => $value){
4.         if($value<0)
5.             {
6.                 return back();
7.             }
8.         $i=$i+$value;
9.     }
10.    if($i!=100)
11.    {
12.        return redirect('/homeDesainer')->with('status', 'Persentase tidak 100!');
13.    }
14.    foreach ($request->idFitur as $index => $idFit) {
15.        $value = new Memberivalue;
16.        $value->id_fitur=$idFit;
```

17.	\$value->id_desainer=\$request->idDesainer;
18.	\$value->nilai_desainer=\$request->persenFitur[\$index];
19.	\$value->save();
20.	}
21.	return view('thanks');

Kode Sumber 4.3 Fungsi Memberi Persentase Setiap Fitur

4.2.1.1.4 Implementasi Proses Memberi Waktu dan Biaya Fitur

Proses ini akan menangani perekaman waktu dan biaya setiap fitur yang diberikan oleh *developer*. Fungsi yang digunakan untuk proses ini adalah **addMembericost()** pada *programmerController*. Kode program fungsi **addMembericost()** dapat dilihat pada Kode Sumber 4.4.

1.	public function addMembericost(Request \$request){
2.	
3.	foreach (\$request->biaya as \$key => \$value) {
4.	if (\$value<0 \$request->waktu<0)
5.	{
6.	return back();
7.	}
8.	}
9.	
10.	foreach (\$request->idFitur as \$index => \$idFit) {
11.	\$cost = new Membericost;
12.	\$cost->id_fitur=\$idFit;
13.	\$cost->id_programmer=\$request->idProgrammer;
14.	\$cost->biaya=\$request->biaya[\$index];
15.	\$cost->waktu=\$request->waktu[\$index];
16.	
17.	\$cost->save();
18.	}
19.	return view('thanks');
20.	}

Kode Sumber 4.4 Fungsi Memberi Waktu dan Biaya Fitur

4.2.1.1.5 Implementasi Proses Memberi Nilai Fitur

Proses ini akan menangani perekaman nilai setiap fitur yang diberikan oleh pengguna. Fungsi yang digunakan untuk proses ini

adalah **addMenilai()** pada *penggunaController*. Kode program fungsi **addMenilai()** dapat dilihat pada pada Kode Sumber 4.5.

```
1. public function addMenilai(Request $request){
2.     foreach ($request->idFitur as $index => $idFit) {
3.         $value = new Menilai;
4.         $value->id_fitur=$idFit;
5.         $value->id_pengguna=$request->idPengguna;
6.         $value->nilai_pengguna=$request->nilai[$index];
7.         $value->save();
8.     }
9.     return view('thanks');
10. }
```

Kode Sumber 4.5 Fungsi Memberi Nilai Fitur

4.2.1.2 Implementasi Perhitungan Nilai Value

Pengimplementasian perhitungan untuk mendapatkan nilai *value* dari jawaban pengguna dan desainer akan dijelaskan pada beberapa kode sumber yang terdapat pada subbab ini. Pengimplementasian nilai tingkat kepentingan AHP menjadi matriks awal dapat dilihat pada Kode Sumber 4.6. Normalisasi dari matriks awal diimplementasikan seperti yang dapat dilihat pada Kode Sumber 4.7. Perhitungan *EigenValue* dari hasil normalisasi matriks diimplementasikan seperti yang terdapat pada Kode Sumber 4.8.

```
1. for($i=0 ; $i<$jumlah ; $i++)
2. {
3.     $arraySum[$i]=0;
4.     $normalSum[$i]=0;
5.     $eigenVal[$i]=0;
6.     $idv[$i]=$isi[$i]->id_fitur;
7.     for($j=0 ; $j<$jumlah ; $j++)
8.     {
9.         if($isi[$j]->valueahp > $isi[$i]->valueahp)
10.        {
11.            $ahpAwal[$i][$j]=$isi[$j]->valueahp - $isi[$i]->valueahp + 1;
12.        }
13.        else
14.        {
15.            $ahpAwal[$i][$j]=1/($isi[$i]->valueahp - $isi[$j]->valueahp + 1);
```

16.	}
17.	\$arraySum[\$i]=\$arraySum[\$i]+\$ahpAwal[\$i][\$j];
18.	}
19.	}

Kode Sumber 4.6 Pengubahan Nilai Kepentingan Menjadi Matriks Awal

1.	for (\$i=0 ; \$i<\$jumlah ; \$i++)
2.	{
3.	for (\$j=0 ; \$j<\$jumlah ; \$j++)
4.	{
5.	\$normal[\$i][\$j]=\$ahpAwal[\$i][\$j]/\$arraySum[\$i];
6.	\$normalSum[\$j]=\$normalSum[\$j]+\$normal[\$i][\$j];
7.	}
8.	}

Kode Sumber 4.7 Normalisasi Matriks

1.	\$jumlahNormal=array_sum(\$normalSum);
2.	for (\$i=0 ; \$i<\$jumlah ; \$i++)
3.	{
4.	\$eigenVal[\$i]=\$normalSum[\$i]/\$jumlahNormal;
5.	}

Kode Sumber 4.8 Perhitungan *Eigen Value*

4.2.1.3 Implementasi Perhitungan Nilai *Cost*

Pengimplementasian perhitungan untuk mendapatkan nilai *cost* dari waktu dan biaya per fitur akan dijelaskan pada beberapa kode sumber yang terdapat pada subbab ini. Pengimplementasian nilai tingkat kepentingan AHP menjadi matriks awal dapat dilihat pada Kode Sumber 4.9. Normalisasi dari matriks awal diimplementasikan seperti yang dapat dilihat pada Kode Sumber 4.10. Perhitungan *Eigen Value* dari hasil normalisasi matriks diimplementasikan seperti yang terdapat pada Kode Sumber 4.11.

1.	for (\$i=0 ; \$i<\$jumlah ; \$i++)
2.	{
3.	\$arraySum[\$i]=0;

```

4. $normalSum[$i]=0;
5. $eigenVal[$i]=0;
6. $idv[$i]=$isi[$i]->id_fitur;
7. for($j=0 ; $j<$jumlah ; $j++)
8. {
9.     if($isi[$j]->costahp > $isi[$i]-> costahp){
10.         $ahpAwal[$i][$j]=$isi[$j]-> costahp - $isi[$i]-> costahp + 1;
11.     }
12.     else {
13.         $ahpAwal[$i][$j]=1/($isi[$i]-> costahp - $isi[$j]-> costahp + 1);
14.     }
15.     $arraySum[$i]=$arraySum[$i]+$ahpAwal[$i][$j];
16. }
17. }

```

Kode Sumber 4.9 Pengubahan Nilai Kepentingan Menjadi Matriks Awal

```

1. for($i=0 ; $i<$jumlah ; $i++){
2.     for($j=0 ; $j<$jumlah ; $j++) {
3.         $normal[$i][$j]=$ahpAwal[$i][$j]/$arraySum[$i];
4.         $normalSum[$j]=$normalSum[$j]+$normal[$i][$j];
5.     }
6. }

```

Kode Sumber 4.10 Normalisasi Matriks

```

1. $jumlahNormal=array_sum($normalSum);
2. for($i=0 ; $i<$jumlah ; $i++)
3. {
4.     $eigenVal[$i]=$normalSum[$i]/$jumlahNormal;
5. }

```

Kode Sumber 4.11 Perhitungan *EigenValue*

4.2.1.4 Implementasi Pengkategorian Setiap Fitur

Pengimplementasian pemberian kategori untuk setiap fitur dijelaskan pada subbab ini. Hasil prioritas akan dibagi menjadi 4 kategori yaitu:

1. Fitur dengan *value* tinggi dan *cost* rendah
2. Fitur dengan *value* tinggi dan *cost* tinggi
3. Fitur dengan *value* rendah dan *cost* rendah
4. Fitur dengan *value* rendah dan *cost* tinggi

Nilai *value* yang dikategorikan sebagai *value* yang tinggi merupakan nilai yang lebih besar atau sama dengan daripada rata-rata *value*. Kategori *value* rendah adalah ketika nilai *value* lebih kecil dari rata-rata *value*. Sedangkan untuk nilai *cost*, dikategorikan sebagai *cost* yang tinggi apabila *cost* lebih besar daripada rata-ratanya. Untuk kategori *cost* rendah adalah ketika *cost* lebih kecil atau sama dengan rata-ratanya. Pemberian kategori untuk setiap fitur diimplementasikan pada Kode Sumber 4.12.

```

1.  foreach ($aplikasi as $index => $app)
2.  {
3.      $midValue=$midValue + $app->valueahp;
4.      $midCost=$midCost + $app->costahp;
5.  }
6.  $midCost = $midCost/count($aplikasi);;
7.  $midValue = $midValue/count($aplikasi);;
8.
9.  foreach ($aplikasi as $index => $app)
10. {
11.     if($app->valueahp >= $midValue && $app->costahp <= $midCost)
12.     {
13.         DB::table('fitur')->where('id_fitur', '=', $app->id_fitur)-
14.         >update(['group' => 1]);
15.     }
16.     elseif ($app->valueahp >= $midValue && $app->costahp > $midCost)
17.     {
18.         DB::table('fitur')->where('id_fitur', '=', $app->id_fitur)-
19.         >update(['group' => 2]);
20.     }
21.     elseif ($app->valueahp < $midValue && $app->costahp <= $midCost)
22.     {
23.         DB::table('fitur')->where('id_fitur', '=', $app->id_fitur)-
24.         >update(['group' => 3]);
25.     }
26.     elseif ($app->valueahp < $midValue && $app->costahp > $midCost)
27.     {
28.         DB::table('fitur')->where('id_fitur', '=', $app->id_fitur)-
29.         >update(['group' => 4]);
30.     }
31. }

```

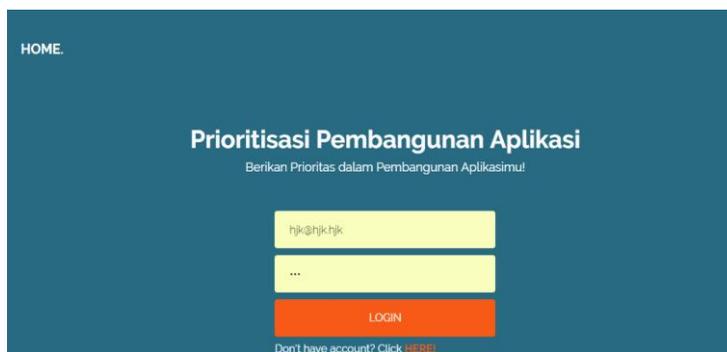
Kode Sumber 4.12 Pengkategorian Hasil Prioritas

4.3 Implementasi Tampilan Antarmuka

Pada subbab ini akan dijelaskan tentang pengimplementasian antarmuka sebagai pendukung proses-proses yang bersangkutan dengan metode yang telah dirancang pada Bab 3. Antarmuka dari sistem ini menjadi bagian terluar sekaligus yang berinteraksi langsung dengan pengguna.

4.3.1.1 Implementasi Antarmuka Halaman Awal *Product Owner*

Halaman ini merupakan halaman yang pertama kali akan diakses oleh *Product Owner*, pada halaman ini *Product Owner* harus melakukan *Login* dengan memberikan *email* dan *password*. Selain itu, pada halaman ini juga dapat dilakukan registrasi sebagai *Product Owner* yang baru dengan cara menekan tombol **HERE**. Implementasi halaman ini dapat dilihat pada Gambar 4.5.

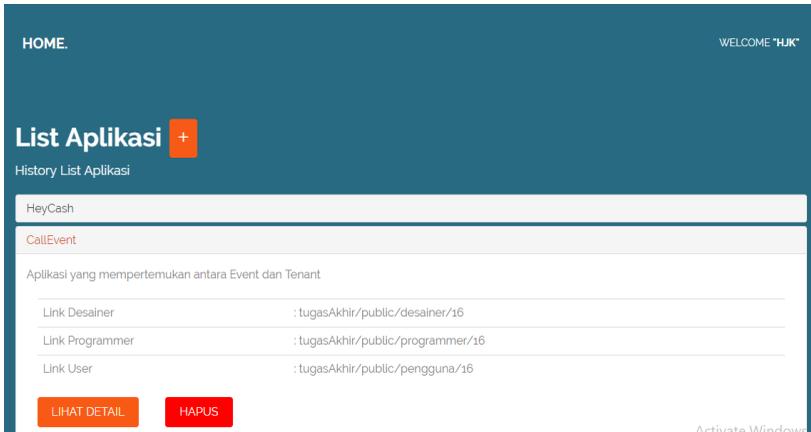


Gambar 4.5 Halaman Awal *Product Owner*

4.3.1.2 Implementasi Antarmuka Halaman Utama *Product Owner*

Halaman utama *Product Owner* ini terdapat informasi mengenai daftar aplikasi yang sudah terekam pada *database*. *Product Owner* dapat melihat detail dari aplikasi dengan menekan pada judul aplikasi yang ditampilkan. Selain itu, aplikasi juga dapat ditambahkan dengan menekan tombol + yang ada pada halaman ini.

Melihat daftar fitur dan detail aplikasi lainnya dapat dilakukan dengan menekan tombol lihat detail yang terdapat pada daftar aplikasi. Penghapusan aplikasi juga dilakukan pada halaman yang sama dengan menekan tombol hapus pada daftar aplikasi yang ada. Implementasi halaman ini dapat dilihat pada Gambar 4.6.



Gambar 4.6 Halaman Utama *Product Owner*

4.3.1.3 Implementasi Antarmuka Menambah Data Aplikasi

Menambah data aplikasi dilakukan pada halaman utama *Product Owner* dengan menekan tombol + yang terdapat pada halaman tersebut, lalu sistem akan menampilkan *form* untuk menambah data aplikasi secara *pop-up*. Pada *form* tersebut terdapat tombol x untuk menggagalkan proses menambah aplikasi, tombol + untuk menambah daftar fitur, tombol dengan *icon* tempat sampah menghapus daftar fitur yang ingin diisikan serta tombol *SAVE* yang digunakan untuk menyimpan data aplikasi baru yang diisikan. Implementasi halaman ini dapat dilihat pada Gambar 4.7.

Gambar 4.7 Halaman Menambah Data Aplikasi

4.3.1.4 Implementasi Antarmuka Halaman Detail Aplikasi

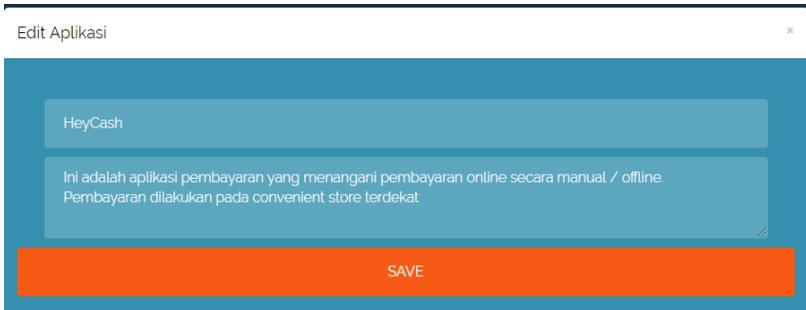
Halaman ini menampilkan detail informasi mengenai aplikasi yang dipilih. Detail informasi tersebut yaitu nama dan deskripsi fitur-fitur yang ada. Pada halaman ini dapat dilakukan perubahan informasi aplikasi dan penambahan atau perubahan data fitur. Implementasi halaman ini dapat dilihat pada Gambar 4.8.

Gambar 4.8 Halaman Menambah Detail Aplikasi

4.3.1.5 Implementasi Antarmuka Mengubah Data Aplikasi

Pada halaman detail aplikasi dapat dilakukan perubahan informasi atau detail dari aplikasi tersebut. Jika tombol *edit* aplikasi

ditekan maka sistem akan menampilkan *form edit* aplikasi dengan menampilkan data sebelumnya yang ada. Setelah data diubah dan dengan menekan tombol *Save* pada *form* tersebut, sistem akan mengubah data aplikasi yang ada sesuai data baru yang diisikan. Implementasi halaman ini dapat dilihat pada Gambar 4.9.



Gambar 4.9 Halaman Mengubah Data Aplikasi

4.3.1.6 Implementasi Antarmuka Menambah Data Fitur

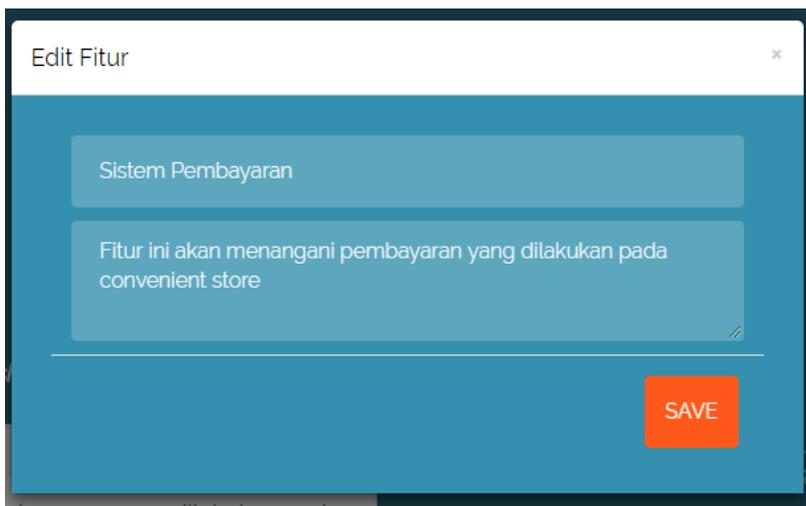
Pada halaman detail aplikasi dapat dilakukan penambahan data fitur untuk aplikasi tersebut. Jika tombol + ditekan maka sistem akan menampilkan *form* tambah fitur yang dapat diisikan data fitur baru. Setelah data baru diisikan dengan menekan tombol *Save* pada *form* tersebut, sistem akan menambahkan data fitur yang baru. Implementasi halaman ini dapat dilihat pada Gambar 4.10.



Gambar 4.10 Halaman Menambah Data Fitur

4.3.1.7 Implementasi Antarmuka Mengubah Data Fitur

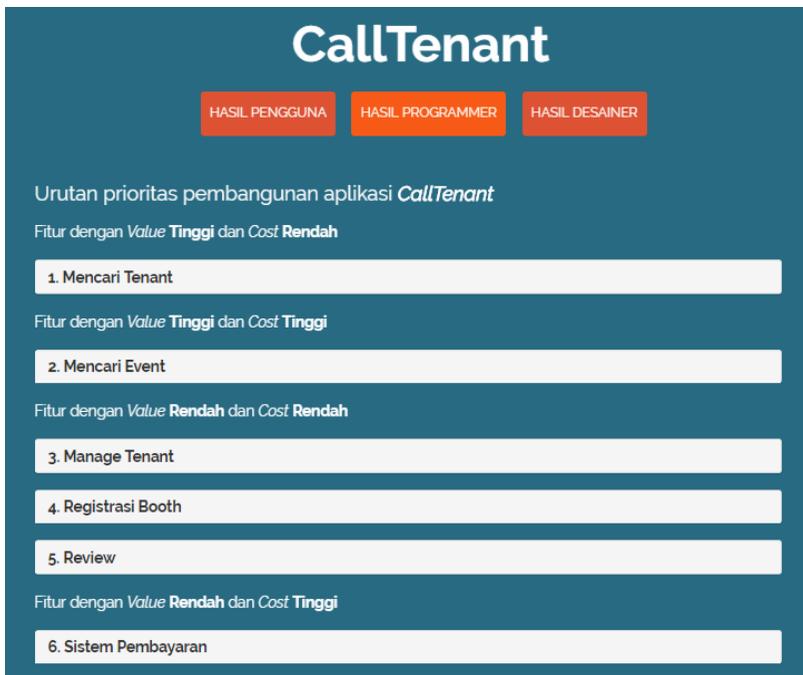
Pada halaman detail aplikasi dapat dilakukan perubahan informasi atau detail dari fitur dari aplikasi tersebut. Jika tombol *Edit* Fitur ditekan maka sistem akan menampilkan *form Edit* Fitur dengan menampilkan data sebelumnya. Setelah data diubah dan dengan menekan tombol *Save* pada *form* tersebut, sistem akan mengubah data fitur yang ada sesuai data baru yang diisikan. Implementasi halaman ini dapat dilihat pada Gambar 4.11.



Gambar 4.11 Halaman Mengubah Data Fitur

4.3.1.8 Implementasi Antarmuka Halaman Hasil Prioritas

Halaman hasil prioritas ini menampilkan hasil berupa daftar fitur kepada *Product Owner*. Pada halaman ini juga terdapat beberapa tombol yang digunakan untuk melihat detail jawaban dari desainer, *developer*, dan pengguna. Selain itu, jika nama fitur ditekan sistem akan menunjukkan detail dari fitur tersebut, yaitu biaya dan lama waktu pengerjaan aplikasi tersebut serta ketergantungan fitur tersebut terhadap fitur lainnya pada aplikasi. Implementasi halaman ini dapat dilihat pada Gambar 4.12.



Gambar 4.12 Halaman Hasil Prioritas

4.3.1.9 Implementasi Antarmuka Detail Hasil Pengguna

Pada halaman hasil prioritas terdapat akses untuk melihat detail hasil jawaban dari pengguna. Jika tombol HASIL PENGGUNA seperti pada Gambar 4.12 ditekan, sistem akan menampilkan *pop-up* berupa informasi detail jawaban pengguna. Jawaban tersebut terdiri dari akumulasi atau rekapan seluruh jawaban pengguna yang telah memberikannya pada aplikasi tersebut, serta detail jawaban per individu dari pengguna. Implementasi halaman ini dapat dilihat pada Gambar 4.13.

Detail Hasil Pengguna

- Akumulasi jawaban Pengguna -

Nama Fitur	Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
Manage Tenant	3	2	0	0
Mencari Event	4	1	0	0
Mencari Tenant	4	1	0	0
Registrasi Booth	4	1	0	0
Review	2	3	0	0
Sistem Pembayaran	3	2	0	0

- Hasil jawaban per Pengguna -

- Kezia Ayuningtyas (keziaay@gmail.com)
- Habibur Rahman (mhabiburr17@gmail.com)

Gambar 4.13 Halaman Detail Hasil Pengguna

4.3.1.10 Implementasi Antarmuka Detail Hasil Desainer

Pada halaman hasil prioritas terdapat akses untuk melihat detail hasil jawaban dari desainer. Jika tombol HASIL DESAINER seperti pada Gambar 4.12 ditekan, sistem akan menampilkan *pop-up* berupa informasi detail jawaban desainer. Jawaban tersebut terdiri dari rata-rata jawaban desainer serta detail jawaban per individu desainer yang mengisikan pendapat mereka. Implementasi halaman ini dapat dilihat pada Gambar 4.14.

Detail Hasil Desainer

- Akumulasi jawaban Desainer -

Nama Fitur	Persentase
Manage Tenant	20 %
Mencari Event	22,5 %
Mencari Tenant	30 %
Registrasi Booth	17,5 %
Review	2,5 %
Sistem Pembayaran	7,5 %

Gambar 4.14 Halaman Detail Hasil Desainer

4.3.1.11 Implementasi Antarmuka Detail Hasil *Developer*

Pada halaman hasil prioritas terdapat akses untuk melihat detail hasil jawaban dari *developer*. Jika tombol HASIL DEVELOPER seperti pada Gambar 4.12 ditekan, sistem akan menampilkan *pop-up* yaitu informasi detail jawaban *developer*. Jawaban tersebut terdiri dari biaya dan waktu pengerjaan untuk setiap fitur. Selain jawaban dari *developer*, informasi mengenai data *developer* juga tercantum pada halaman ini. Implementasi halaman ini dapat dilihat pada Gambar 4.15.



Nama Fitur	Waktu Pengerjaan	Biaya
Mencari Event	10 hari	Rp 2000000
Mencari Tenant	10 hari	Rp 2000000
Sistem Pembayaran	15 hari	Rp 15000000
Review	5 hari	Rp 1000000
Registrasi Booth	5 hari	Rp 1000000
Manage Tenant	7 hari	Rp 1000000

Gambar 4.15 Halaman Detail Hasil *Developer*

4.3.1.12 Implementasi Antarmuka Halaman Awal Pengguna

Halaman awal pengguna adalah halaman yang memiliki akses untuk memberikan nilai dari aplikasi sebagai pengguna. Halaman ini memiliki *form* yang harus diisi oleh pengguna berupa nama dan *email*. Data di sini akan dilihat dan disamakan dengan data dari *database* untuk mengetahui apakah pengguna yang sama telah memberikan nilai pada aplikasi yang sama atau belum. Implementasi halaman ini dapat dilihat pada Gambar 4.16.



Gambar 4.16 Halaman Awal Pengguna

4.3.1.13 Implementasi Antarmuka Halaman Menilai

Pada halaman ini pengguna dapat memberikan nilai pada setiap fitur yang terdapat pada aplikasi. Halaman ini menampilkan detail dari aplikasi dan fiturnya serta pilihan jawaban yang bisa dipilih oleh pengguna. Implementasi halaman ini dapat dilihat pada Gambar 4.17.



Gambar 4.17 Halaman Menilai

4.3.1.14 Implementasi Antarmuka Halaman Awal Desainer

Halaman awal desainer adalah halaman yang memiliki akses untuk memberikan persentase dari aplikasi sebagai desainer. Halaman ini memiliki *form* yang harus diisi oleh desainer seperti nama dan *email*. Data di sini akan dilihat dan disamakan dengan data

dari *database* untuk mengetahui apakah desainer yang sama telah memberikan persentase pada aplikasi yang sama. Implementasi halaman ini dapat dilihat pada Gambar 4.18.



Gambar 4.18 Halaman Awal Desainer

4.3.1.15 Implementasi Antarmuka Halaman Memberi Persentase

Pada halaman ini desainer dapat memberikan persentase pada setiap fitur yang terdapat pada aplikasi. Halaman ini menampilkan detail dari aplikasi dan fiturnya serta tempat untuk mengisikan persentase bagi desainer. Implementasi halaman ini dapat dilihat pada Gambar 4.19.



Gambar 4.19 Halaman Memberi Persentase

4.3.1.16 Implementasi Antarmuka Halaman Awal *Developer*

Halaman awal *developer* adalah halaman yang memiliki akses untuk memberikan biaya dan waktu untuk setiap fitur. Halaman ini memiliki *form* yang harus diisi oleh *developer* berupa biaya dan waktu. Data di sini akan dilihat dan disamakan dengan data dari *database* untuk mengetahui apakah sudah ada data *cost* yang tersimpan untuk aplikasi yang sama. Implementasi halaman ini dapat dilihat pada Gambar 4.20.



Gambar 4.20 Halaman Awal *Developer*

4.3.1.17 Implementasi Antarmuka Halaman Memberi *Cost*

Pada halaman ini, setelah mendapatkan akses sebagai *developer*, maka *user* dapat memberikan *cost* untuk setiap fitur aplikasi. Halaman ini akan menampilkan detail informasi terkait aplikasi dan fiturnya. *Developer* diminta untuk mengisi data biaya dan waktu yang dibutuhkan untuk membangun fitur tersebut. Implementasi halaman ini dapat dilihat pada Gambar 4.21.

Detail Hasil Programmer		
Nama : Marthin SP		
Email : marthinpsrb@gmail.com		
Nama Fitur	Waktu Pengerjaan	Biaya
Mencari Event	10 hari	Rp 2000000
Mencari Tenant	10 hari	Rp 2000000
Sistem Pembayaran	15 hari	Rp 15000000
Review	5 hari	Rp 1000000
Registrasi Booth	5 hari	Rp 1000000
Manage Tenant	7 hari	Rp 1000000

Gambar 4.21 Halaman Memberi Cost

4.3.1.18 Implementasi Antarmuka Halaman Terima Kasih

Halaman ini bertujuan untuk memberikan informasi bahwa data yang diisikan sudah terekam dalam *database*. Halaman ini hanya berisikan kalimat yang menyatakan data tersimpan dan sebuah tombol OK untuk kembali ke menu awal aplikasi. Implementasi halaman ini dapat dilihat pada Gambar 4.22.



Gambar 4.22 Halaman Terima Kasih

4.4 Implementasi Perangkat Lunak

Pada subbab ini akan dijelaskan mengenai implementasi proses pendukung atau penyempurnaan dari implementasi proses-proses yang menunjang metode pengerjaan Tugas Akhir ini, seperti yang telah dijelaskan pada subbab 4.2.1. Subbab ini akan menjelaskan proses tersebut dalam bentuk kode. Proses penunjang yang dibahas adalah proses-proses yang sudah dirancang pada Bab 3.

4.4.1 Implementasi Proses Mengubah Aplikasi

Proses ini dilakukan oleh *Product Owner* untuk mengubah data aplikasi yang sudah ada. Implementasi proses ini adalah menggunakan fungsi **editAplikasi()** yang akan mengubah data yang sudah ada pada *database*. Untuk menjalankan fungsi ini, *Product Owner* harus menekan tombol *edit* aplikasi, lalu mengisikan data terbaru aplikasi. Setelah tombol *Save* ditekan, sistem akan menuju fungsi **editAplikasi()** yang berada pada *appController*. Fungsi **editAplikasi()** ini akan mengubah data yang sudah ada pada *database*, fungsi ini dapat dilihat pada Kode Sumber 4.13.

```
1. public function editAplikasi(Request $request)
2. {
3.     $aplikasi = Aplikasi::find($request->idApp);
4.     $aplikasi->nama_aplikasi = $request->namaApp;
5.     $aplikasi->deskripsi_aplikasi = $request->desApp;
6.     $aplikasi->save();
7.     return back();
8. }
```

Kode Sumber 4.13 Fungsi Mengubah Aplikasi

4.4.2 Implementasi Proses Menghapus Aplikasi

Proses ini dilakukan oleh *Product Owner* untuk menghapus data aplikasi yang ada pada *database*. Implementasi proses ini menggunakan fungsi **hapusApp()** yang akan menghapus data yang sudah ada pada *database*. Untuk menjalankan fungsi ini, *Product Owner* harus menekan tombol hapus aplikasi, sistem akan menuju fungsi **hapusApp()** yang berada pada *appController*. Fungsi

hapusApp() ini akan menghapus data aplikasi, fungsi ini dapat dilihat pada pada Kode Sumber 4.14.

1.	public function hapusApp(Request \$request,\$id)
2.	{
3.	DB::table('aplikasi')->where('id_aplikasi', '=', \$id)->delete();
4.	return back();
5.	}

Kode Sumber 4.14 Fungsi Menghapus Aplikasi

4.4.3 Implementasi Proses Mengubah Fitur

Proses ini dilakukan oleh *Product Owner* untuk mengubah data fitur yang sudah ada. Implementasi proses ini menggunakan fungsi **editFitur()** yang akan mengubah data yang sudah ada pada *database*. Untuk menjalankan fungsi ini, *Product Owner* harus menekan tombol *Edit* Fitur, lalu mengisikan data terbaru fitur. Setelah tombol *Save* ditekan, sistem akan menuju fungsi **editFitur()** yang berada pada *appController*. Fungsi **editFitur()** ini akan mengubah data fitur, fungsi ini dapat dilihat pada pada Kode Sumber 4.15.

1.	public function editFitur(Request \$request,\$id)
2.	{
3.	\$fitur = Fitur ::find(\$id);
4.	\$fitur->nama_fitur = \$request->namaFit;
5.	\$fitur->deskripsi_fitur = \$request->desFit;
6.	\$fitur-> save ();
7.	\$idApp = DB::select("select id_aplikasi from fitur where id_fitur=\$id");
8.	return back();
9.	}

Kode Sumber 4.15 Fungsi Mengubah Fitur

4.4.4 Implementasi Proses Menghapus Fitur

Proses ini dilakukan oleh *Product Owner* untuk menghapus data fitur yang ada pada *database*. Implementasi proses ini menggunakan fungsi **hapusFitur()** yang akan menghapus data yang sudah ada pada *database*. Untuk menjalankan fungsi ini, *Product Owner* harus menekan tombol hapus fitur, sistem akan menuju

fungsi **hapusFitur()** yang berada pada *appController*. Fungsi **hapusFitur()** ini akan menghapus data dan fungsi ini dapat dilihat pada Kode Sumber 4.16.

1.	public function hapusFitur(Request \$request,\$id){
2.	DB::table('fitur')->where('id_fitur', '=', \$id)->delete();
3.	return back();}

Kode Sumber 4.16 Fungsi Menghapus Fitur

4.4.5 Implementasi Proses Perhitungan *Value* Menggunakan AHP

Proses ini adalah perhitungan yang dilakukan untuk mendapatkan nilai *value*. Perhitungan AHP ini dilakukan oleh fungsi **hitungahpV()** pada *appController*. Fungsi ini akan mengambil data yang dibutuhkan, berupa persentase dari desainer dan nilai dari pengguna. Perhitungan ini akan dilakukan ketika halaman hasil prioritas diakses. Perhitungan AHP untuk mendapatkan nilai *value* dapat dilihat pada Kode Sumber 4.17.

1.	public function hitungahpV(\$id){
2.	\$sisi = DB::select("select f.id_fitur as IDF ,(sum(m.nilai_pengguna)/count(m.id_penilai))/ (SELECT SUM(m.nilai_pengguna)/(COUNT(f.id_fitur)/ (SELECT COUNT(id_fitur) FROM fitur WHERE id_aplikasi=\$id)) FROM menilai m, fitur f where m.id_fitur=f.id_fitur AND f.id_aplikasi=\$id)*100*0.658 + sum(mv.nilai_desainer)/count(mv.id_desainer)*0.342 as VALAHP from menilai m, fitur f, memberivalue mv where f.id_fitur=m.id_fitur AND f.id_aplikasi=\$id AND mv.id_fitur=f.id_fitur AND mv.id_fitur=m.id_fitur GROUP BY f.id_fitur ORDER BY VALAHP DESC");
3.	\$jumlah= count (\$sisi);
4.	foreach (\$sisi as \$index => \$nama) {
5.	DB::table('fitur')
6.	->where('id_fitur', \$sisi[\$index]->IDF)
7.	->update(['valueahp' => round(10-(((\$index+1)/\$jumlah*9))]);
8.	}
9.	\$sisi = DB:: select("select * from fitur where id_aplikasi=\$id");
10.	\$jumlah = count(\$sisi);
11.	\$ahpAwal=array();
12.	\$arraySum = array();
13.	\$normalSum=array();

```

14. $eigenVal=array();
15. $idv=array();
16. for($i=0 ; $i<$jumlah ; $i++)
17. {
18.     $arraySum[$i]=0;
19.     $normalSum[$i]=0;
20.     $eigenVal[$i]=0;
21.     $idv[$i]=$isi[$i]->id_fitur;
22.     for($j=0 ; $j<$jumlah ; $j++)
23.     {
24.         if($isi[$j]->valueahp > $isi[$i]->valueahp)
25.         {
26.             $ahpAwal[$i][$j]=$isi[$j]->valueahp - $isi[$i]->valueahp + 1;
27.         }
28.         else
29.         {
30.             $ahpAwal[$i][$j]=1/($isi[$i]->valueahp - $isi[$j]->valueahp + 1);
31.         }
32.         $arraySum[$i]=$arraySum[$i]+$ahpAwal[$i][$j];
33.     }
34. }
35. $normal=array();
36. for($i=0 ; $i<$jumlah ; $i++)
37. {
38.     for($j=0 ; $j<$jumlah ; $j++)
39.     {
40.         $normal[$i][$j]=$ahpAwal[$i][$j]/$arraySum[$i];
41.         $normalSum[$j]=$normalSum[$j]+$normal[$i][$j];
42.     }
43. }
44. $jumlahNormal=array_sum($normalSum);
45. for($i=0 ; $i<$jumlah ; $i++)
46. {
47.     $eigenVal[$i]=$normalSum[$i]/$jumlahNormal;
48. }
49. foreach ($isi as $index => $nama) {
50.     DB::table('fitur')
51.     ->where('id_fitur', $idv[$index])
52.     ->update(['valueahp' => $eigenVal[$index]]);
53. }
54. }

```

Kode Sumber 4.17 Fungsi Perhitungan *Value* Menggunakan AHP

4.4.6 Implementasi Proses Menghitung Cost Menggunakan AHP

Proses ini adalah perhitungan yang dilakukan untuk mendapatkan nilai *cost*. Perhitungan AHP ini dilakukan oleh fungsi **hitungahpC()** pada *appController*. Fungsi ini akan mengambil data yang dibutuhkan dari *cost* yang diberikan oleh *developer*. Perhitungan ini akan dilakukan ketika halaman hasil prioritas diakses. Perhitungan AHP untuk mendapatkan nilai *cost* dapat dilihat pada pada Kode Sumber 4.18.

```
1. public function hitungahpC($id){
2.     $sisi = DB::select("select f.id_fitur as IDF, mc.biaya, mc.waktu,
   mc.biaya/mc.waktu as COSTAHP FROM membericost mc, fitur f
   WHERE f.id_fitur=mc.id_fitur AND f.id_aplikasi=$id ORDER BY
   COSTAHP ASC");
   $jumlah = count($sisi);
   foreach ($sisi as $index => $nama) {
       DB::table('fitur')
       ->where('id_fitur', $sisi[$index]->IDF)
       ->update(['costahp' => round(($index+1)/$jumlah*9)]);
3.   }
4.
5.   $sisi = DB::select("select * from fitur where id_aplikasi=$id");
6.   $jumlah = count($sisi);
7.   $sahpAwal=array();
8.   $arraySum = array();
9.   $normalSum=array();
10.  $eigenVal=array();
11.  $sidv=array();
12.
13.  for($i=0 ; $i<$jumlah ; $i++)
14.  {
15.      $arraySum[$i]=0;
16.      $normalSum[$i]=0;
17.      $eigenVal[$i]=0;
18.      $sidv[$i]=$sisi[$i]->id_fitur;
19.      for($j=0 ; $j<$jumlah ; $j++)
20.      {
21.          if($sisi[$j]->costahp > $sisi[$i]->costahp)
22.          {
23.              $sahpAwal[$i][$j]=$sisi[$j]->costahp - $sisi[$i]->costahp + 1;
```

```

24.     }
25.     else
26.     {
27.         $ahpAwal[$i][$j]=1/($isi[$i]->costahp - $isi[$j]->costahp + 1);
28.     }
29.     $arraySum[$i]=$arraySum[$i]+$ahpAwal[$i][$j];
30.     }
31. }
32.
33. $normal=array();
34. for($i=0 ; $i<$jumlah ; $i++)
35. {
36.     for($j=0 ; $j<$jumlah ; $j++)
37.     {
38.         $normal[$i][$j]=$ahpAwal[$i][$j]/$arraySum[$i];
39.         $normalSum[$j]=$normalSum[$j]+$normal[$i][$j];
40.     }
41. }
42. $jumlahNormal=array_sum($normalSum);
43. for($i=0 ; $i<$jumlah ; $i++)
44. {
45.     $eigenVal[$i]=$normalSum[$i]/$jumlahNormal;
46. }
47.
48. foreach ($isi as $index => $nama) {
49.     DB::table('fitur')
50.     ->where('id_fitur', $idv[$index])
51.     ->update(['costahp' => $eigenVal[$index]]);
52. }
53.
54. }

```

Kode Sumber 4.18 Fungsi Menghitung Cost Menggunakan AHP

4.4.7 Implementasi Proses Melihat Hasil Prioritas Aplikasi

Proses ini akan menampilkan hasil prioritas dari aplikasi serta detail jawaban dari pengguna, desainer, dan *developer*. Fungsi yang akan melakukan proses ini adalah fungsi **hasilPrioritas()** yang terdapat pada *appController*. Fungsi **hasilPrioritas()** akan memanggil fungsi lainnya yaitu **hitungahpV()** dan **hitungahpC()** yang digunakan untuk menghitung nilai AHP dari setiap fitur. Fungsi ini dapat dilihat pada pada Kode Sumber 4.19.

```

1. public function hasilPrioritas(Request $request,$id){
2.     $this->hitungahpC($id);
3.     $this->hitungahpV($id);
4.
5.     $midValue=0;
6.     $totVal=0;
7.     $midCost=0;
8.     $totCost=0;
9.
10.    $aplikasi = DB::select("select * from aplikasi a, fitur f where
a.id_aplikasi=f.id_aplikasi and f.id_aplikasi=$id ORDER BY f.valueahp
DESC, f.costahp");
11.
12.    foreach ($aplikasi as $index => $app) {
13.        $midValue=$midValue + $app->valueahp;
14.        $midCost=$midCost + $app->costahp;
15.    }
16.    $midCost = $midCost/count($aplikasi);;
17.    $midValue = $midValue/count($aplikasi);;
18.
19.    foreach ($aplikasi as $index => $app) {
20.        if($app->valueahp >= $midValue && $app->costahp <= $midCost)
21.        {
22.            DB::table('fitur')->where('id_fitur', '=', $app->id_fitur)-
>update(['group' => 1]);
23.        }
24.        elseif ($app->valueahp >= $midValue && $app->costahp >
$midCost)
25.        {
26.            DB::table('fitur')->where('id_fitur', '=', $app->id_fitur)-
>update(['group' => 2]);
27.        }
28.        elseif ($app->valueahp < $midValue && $app->costahp <=
$midCost)
29.        {
30.            DB::table('fitur')->where('id_fitur', '=', $app->id_fitur)-
>update(['group' => 3]);
31.        }
32.        elseif ($app->valueahp < $midValue && $app->costahp > $midCost)
33.        {
34.            DB::table('fitur')->where('id_fitur', '=', $app->id_fitur)-
>update(['group' => 4]);
35.        }

```

36.	}
37.	
38.	\$aplikasi = DB::select("select * from aplikasi a, fitur f, membericost m
39.	where a.id_aplikasi=f.id_aplikasi and f.id_aplikasi=\$id and m.id_fitur=f.id_fitur ORDER BY f.group ASC, f.valueahp desc, f.costahp asc");
40.	\$ketergantungan = DB::select("select distinct k.id_ketergantungan as
41.	idKetergantungan, k.id_fitur as idFiturUtama, f1.nama_fitur as fiturUtama, k.id_fitur_tergantungan as idFiturBergantung, f2.nama_fitur as fiturBergantung from ketergantungan k, fitur f1, fitur f2 where k.id_fitur=f1.id_fitur and k.id_fitur_tergantungan=f2.id_fitur and f1.id_aplikasi=\$id and f2.id_aplikasi=\$id");
42.	\$pengguna = DB::select("select m.id_pengguna as id, f.nama_fitur as
43.	nama, m.nilai_pengguna as nilai from fitur f, pengguna p, menilai m WHERE f.id_fitur=m.id_fitur and m.id_pengguna=p.id_pengguna and f.id_aplikasi=\$id");
44.	\$namaPengguna = DB::select("select DISTINCT p.id_pengguna as id, p.nama_pengguna as nama, p.email_pengguna as email from fitur f, pengguna p, menilai m WHERE f.id_fitur=m.id_fitur and m.id_pengguna=p.id_pengguna and f.id_aplikasi=\$id");
45.	\$totalPengguna = DB::select("select DISTINCT f.nama_fitur as
46.	nama_fitur from fitur f, pengguna p, menilai m WHERE
47.	f.id_fitur=m.id_fitur and m.id_pengguna=p.id_pengguna and f.id_aplikasi=\$id GROUP BY f.nama_fitur");
48.	
49.	\$programmer = DB::select("select * from fitur f, programmer p,
50.	membericost m WHERE f.id_fitur=m.id_fitur and m.id_programmer=p.id_programmer and f.id_aplikasi=\$id");
51.	\$desainer = DB::select("select * from fitur f, desainer d, memberivalue
52.	m WHERE f.id_fitur=m.id_fitur and m.id_desainer=d.id_desainer and f.id_aplikasi=\$id");
53.	return view('hasilPrioritas', ['aplikasi'=>\$aplikasi,
54.	'pengguna'=>\$pengguna, 'ketergantungan'=>\$ketergantungan, 'programmer'=>\$programmer, 'totalPengguna'=>\$totalPengguna, 'namaPengguna'=>\$namaPengguna, 'desainer'=>\$desainer]);
55.	}

Kode Sumber 4.19 Fungsi Melihat Hasil Prioritas Aplikasi

4.4.8 Implementasi Proses Melihat Jawaban Desainer

Proses ini akan menampilkan jawaban desainer secara keseluruhan. Informasi yang akan diberikan adalah rata-rata persentase setiap fitur berdasarkan pendapat desainer. Selain itu detail jawaban tiap desainer pun dapat diakses pada halaman yang sama. Untuk implementasinya dilakukan pada halaman yang sama dengan halaman hasil prioritas, dan fungsi yang dibutuhkan untuk proses melihat jawaban desainer dapat dilihat pada Kode Sumber 4.19.

4.4.9 Implementasi Proses Melihat Jawaban *Developer*

Proses ini akan menampilkan jawaban *developer* secara keseluruhan. Informasi yang akan diberikan adalah biaya dan waktu yang dibutuhkan untuk setiap fitur. Selain itu data berupa nama dan *email developer* juga turut dilampirkan. Untuk implementasinya dilakukan pada halaman yang sama dengan halaman hasil prioritas, dan fungsi yang dibutuhkan untuk proses melihat jawaban *developer* dapat dilihat pada Kode Sumber 4.19.

4.4.10 Implementasi Proses Melihat Jawaban Pengguna

Proses ini akan menampilkan jawaban pengguna secara keseluruhan. Informasi yang akan diberikan adalah akumulasi total dari jawaban pengguna setiap fitur. Selain itu, detail jawaban per pengguna juga dapat diakses pada halaman yang sama. Untuk implementasinya dilakukan pada halaman yang sama dengan halaman hasil prioritas, dan fungsi yang dibutuhkan untuk proses melihat jawaban pengguna dapat pada Kode Sumber 4.19.

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang telah dibangun. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsional. Pengujian fungsionalitas ini mengacu pada kasus penggunaan yang telah dijabarkan pada Bab 3. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1 Lingkungan Pengujian

Lingkungan pengujian aplikasi ini dilakukan pada lingkungan dengan spesifikasi sebagai berikut:

Jenis	: Laptop
Tipe	: Asus A455L
Prosesor	: Intel® Core™ i7-4510U (3.60 GHz)
Memori/RAM	: 4 GB

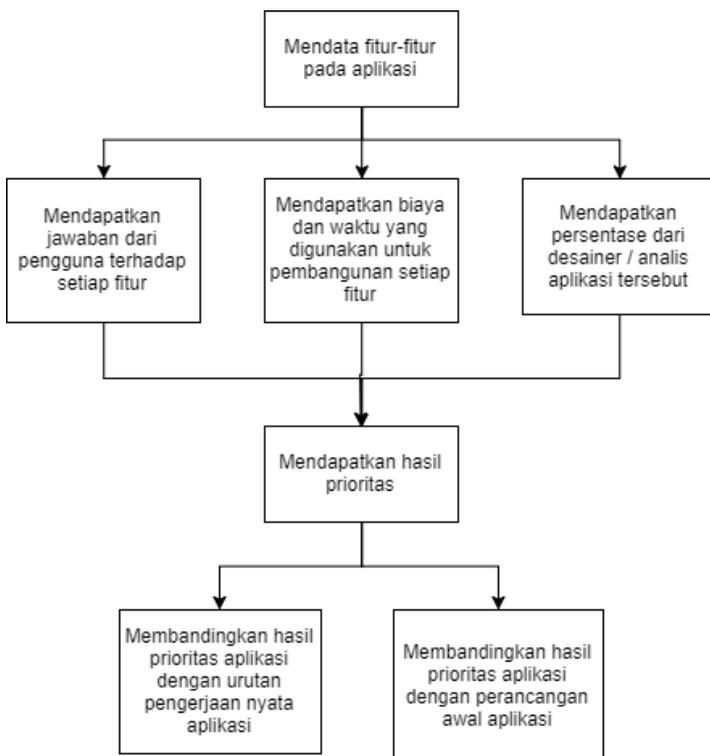
Jenis	: Laptop
Tipe	: HP ProBook 450 G4
Prosesor	: Intel® Core™ i5-6200U (3.60 GHz)
Memori/RAM	: 8 GB

5.2 Skenario Pengujian

Subbab ini menjelaskan mengenai skenario pengujian fungsionalitas yang ada pada aplikasi. Dengan mengacu kepada Bab 3 untuk kebenaran metode dan kebutuhan fungsionalitasnya. Metode yang dilakukan dalam pengujian ini dibagi menjadi dua metode, yaitu:

1. Pengujian kebenaran metode dengan membandingkan prioritas hasil aplikasi Priro dan prioritas hasil dari aplikasi AHP yang sudah ada.
2. Pengujian fungsionalitas aplikasi menggunakan metode *black box*. Metode *black box* merupakan metode di mana pengujian ditekankan pada pola *input* dan *output* sesuai dengan skenario.

Detail langkah pengujian dapat dilihat pada Gambar 5.1.



Gambar 5.1 Langkah Pengujian Perangkat Lunak / Aplikasi

5.2.1 Pengujian Perancangan Suatu Perangkat Lunak dengan Bantuan Aplikasi

Pada pengujian ini dilakukan pada dua buah perangkat lunak dengan nama CallTenant dan WithYou. Pertama-tama hal yang akan dilakukan adalah pendataan fitur-fitur utama dari aplikasi yang ingin dibangun, lalu menanyakan data dari desainer perangkat lunak tersebut mengenai urutan pembangunan dari aplikasi tersebut. Selain itu, waktu dan biaya pengerjaan dari setiap fitur turut ditanyakan. Untuk menyempurnakan perhitungan, dilakukan pengambilan

pendapat atau nilai dari responden yang berkaitan dengan perangkat lunak tersebut. Lebih jelasnya akan dijelaskan pada subbab per perangkat lunak. Langkah-langkah pengerjaan uji coba ini dijelaskan pada Gambar 5.1.

Pada langkah-langkah pengujian dijelaskan bahwa setelah mendapatkan hasil prioritas dari aplikasi urutan pembangunan fitur tersebut akan dibandingkan dengan perancangan oleh desainer dari pembangun perangkat lunak tersebut. Selain itu, hasil prioritas juga dibandingkan dengan kenyataan urutan pembangunan perangkat lunak tersebut.

Kedua perangkat lunak yang diujikan pada aplikasi Priro merupakan sebuah *startup*. Di mana kebanyakan pembangunan *startup* dilakukan secara cepat dengan mengiterasi proses yang dilakukan untuk mendapatkan fitur yang sempurna atau yang diinginkan. Oleh sebab itu, kebanyakan dari *startup* akan memulai pembangunannya dari fitur yang sedikit lalu dengan pengembangannya jika memang ada yang harus ditambahkan. Hal ini juga diterapkan oleh tim pembangun perangkat lunak CallTenant dan WithYou. Oleh karena itu, jumlah fitur dari perangkat lunak yang digunakan dalam uji coba ini berjumlah sedikit.

5.2.1.1 Pengujian Perancangan Aplikasi CallTenant

Subbab ini membahas pengujian yang dilakukan dalam pembangunan sebuah perangkat lunak dengan nama CallTenant. CallTenant merupakan sebuah perangkat lunak yang memiliki tujuan untuk menghubungkan antara penyelenggara acara (*Event Organizer*) dan pemilik usaha yang ingin membuka *booth* pada acara tersebut (*Tenant*) [10].

Pengujian dilakukan dari tahapan awal analisis dan perancangan aplikasi CallTenant. Pengambilan data *stakeholder* dilakukan pada 15 pengguna yang dibagi seperti pada Tabel 5.1. Di mana yang berperan sebagai *Product Owner* adalah orang yang terlibat langsung dalam pembangunan CallTenant. Desainer merupakan orang yang merancang pembangunan CallTenant. *Developer* adalah orang yang melakukan pembangunan aplikasi.

Pengguna di sini merupakan pelaku yang berkaitan langsung dengan aplikasi ini. Pengguna yang dimaksud adalah pemilik usaha yang pernah menjadi *tenant* dalam suatu acara atau seseorang yang pernah menjadi bagian penyelenggara acara.

Tabel 5.1 Demografi Penguji Coba

Nama Peran	Jumlah Penguji Coba
<i>Product Owner</i>	1
Desainer	3
<i>Developer</i>	1
Pengguna	10

Petama-tama *Product Owner* memberikan informasi daftar fitur untuk aplikasi CallTenant. Daftar fitur dari aplikasi tersebut dapat dilihat pada Tabel 5.2.

Tabel 5.2 Daftar Fitur Aplikasi CallTenant

No.	Nama Fitur
1.	Mencari <i>Event</i>
2.	Mencari <i>Tenant</i>
3.	Sistem Pembayaran
4.	<i>Review</i>
5.	Registrasi <i>Booth</i>
6.	<i>Manage Tenant</i>

Hasil jawaban dari pengguna yang memberikan nilai untuk setiap fitur, dapat dilihat pada Tabel 5.3.

Tabel 5.3 Hasil Jawaban Pengguna

Nama Fitur	Sangat Dibutuhkan	Dibutuhkan	Mungkin Dibutuhkan	Tidak Dibutuhkan
Mencari <i>Event</i>	9	1	0	0
Mencari <i>Tenant</i>	9	1	0	0
Sistem Pembayaran	6	4	0	0

<i>Review</i>	2	7	0	1
<i>Registrasi Booth</i>	7	3	0	0
<i>Manage Tenant</i>	7	3	0	0

Hasil jawaban dari desainer yang memberikan persentase untuk setiap fitur, dapat dilihat pada Tabel 5.4.

Tabel 5.4 Hasil Jawaban Desainer

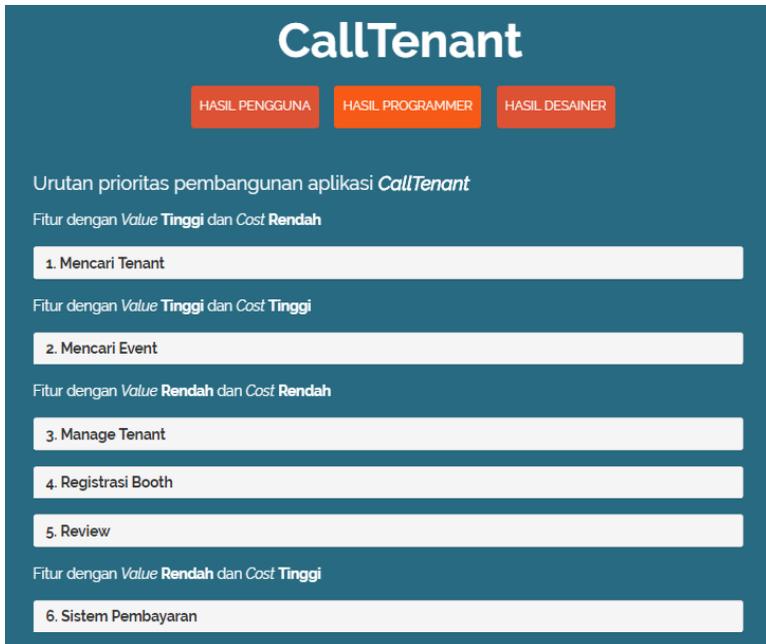
Nama Fitur	Desainer 1	Desainer 2	Desainer 3
<i>Mencari Event</i>	25%	20%	20%
<i>Mencari Tenant</i>	40%	20%	20%
<i>Sistem Pembayaran</i>	10%	5%	15%
<i>Review</i>	0%	5%	10%
<i>Registrasi Booth</i>	10%	25%	20%
<i>Manage Tenant</i>	15%	25%	15%

Hasil jawaban dari *developer* yang memberikan waktu dan biaya yang dibutuhkan untuk setiap fitur, dapat dilihat pada Tabel 5.5.

Tabel 5.5 Hasil Jawaban Developer

Nama Fitur	Waktu	Biaya
<i>Mencari Event</i>	10 hari	Rp 2,000,000
<i>Mencari Tenant</i>	10 hari	Rp 2,000,000
<i>Sistem Pembayaran</i>	15 hari	Rp 15,000,000
<i>Review</i>	5 hari	Rp 1,000,000
<i>Registrasi Booth</i>	5 hari	Rp 1,000,00
<i>Manage Tenant</i>	7 hari	Rp 1,000,00

Hasil prioritas perancangan CallTenant dengan menggunakan aplikasi Priro dapat dilihat pada Gambar 5.2.



Gambar 5.2 Hasil Prioritas CallTenant

5.2.1.2 Pengujian Perancangan Aplikasi WithYou

Subbab ini membahas pengujian yang dilakukan dalam pembangunan sebuah perangkat lunak dengan nama WithYou. WithYou merupakan sebuah perangkat lunak yang memiliki tujuan untuk membantu *bride-to-be* dalam mencari kebutuhannya untuk melangsungkan pernikahan. Dalam aplikasi ini *bride-to-be* mampu mencari vendor dan produk yang diinginkan. Fitur yang membuat perangkat lunak ini berbeda dari yang sudah ada adalah fitur *recommendation budget* [11].

Pengujian dilakukan pada 14 pengguna yang dibagi seperti pada Tabel 5.6. Di mana yang berperan sebagai *Product Owner* adalah orang yang terlibat langsung dalam pembangunan perangkat lunak WithYou. Desainer merupakan orang yang merancang pembangunan WithYou. *Developer* adalah orang yang melakukan

pembangunan aplikasi. Pengguna di sini merupakan pelaku yang berkaitan langsung dengan aplikasi ini. Pengguna yang dimaksud adalah *bride-to-be* atau masyarakat milenial yang berada pada usia cukup untuk menikah (usia 21 hingga 31 tahun).

Tabel 5.6 Demografi Penguji Coba

Nama Peran	Jumlah Penguji Coba
<i>Product Owner</i>	1
Desainer	2
<i>Developer</i>	1
Pengguna	10

Petama-tama *Product Owner* memberikan informasi Daftar fitur untuk aplikasi WithYou. Daftar fitur dari aplikasi tersebut dapat dilihat pada Tabel 5.7.

Tabel 5.7 Daftar Fitur Aplikasi WithYou

No.	Nama Fitur
1.	<i>Calculate Budget</i>
2.	<i>Search Vendor</i>
3.	<i>Recommendation Vendor</i>
4.	<i>Search Product</i>
5.	<i>Recommendation Product</i>
6.	<i>Recommendation Budget / Price</i>
7.	<i>Shopping</i>

Hasil jawaban dari pengguna yang memberikan nilai untuk setiap fitur, dapat dilihat pada Tabel 5.8.

Tabel 5.8 Hasil Jawaban Pengguna

Nama Fitur	Sangat Dibutuhkan	Dibutuhkan	Mungkin Dibutuhkan	Tidak Dibutuhkan
<i>Calculate Budget</i>	8	2	0	0

<i>Search Vendor</i>	6	4	0	0
<i>Recommendation Vendor</i>	4	3	3	0
<i>Search Product</i>	5	3	2	0
<i>Recommendation Product</i>	2	3	5	0
<i>Recommendation Budget / Price</i>	4	3	2	1
<i>Shopping</i>	1	2	4	3

Hasil jawaban dari desainer yang memberikan persentase untuk setiap fitur, dapat dilihat pada Tabel 5.9.

Tabel 5.9 Hasil Jawaban Desainer

Nama Fitur	Desainer 1	Desainer 2
<i>Calculate Budget</i>	25%	37%
<i>Search Vendor</i>	10%	10%
<i>Recommendation Vendor</i>	10%	10%
<i>Search Product</i>	15%	5%
<i>Recommendation Product</i>	10%	5%
<i>Recommendation Budget / Price</i>	25%	30%
<i>Shopping</i>	5%	3%

Hasil jawaban dari *developer* yang memberikan waktu dan biaya yang dibutuhkan untuk setiap fitur, dapat dilihat pada Tabel 5.10.

Tabel 5.10 Hasil Jawaban Developer

Nama Fitur	Waktu	Biaya
<i>Calculate Budget</i>	10 hari	Rp 10.000.000
<i>Search Vendor</i>	7 hari	Rp 3.000.000

<i>Recommendation Vendor</i>	7 hari	Rp 4.000.000
<i>Search Product</i>	5 hari	Rp 2.000.000
<i>Recommendation Product</i>	5 hari	Rp 2.000.000
<i>Recommendation Budget / Price</i>	15 hari	Rp 7.000.000
<i>Shopping</i>	10 hari	Rp 5.000.000

Hasil prioritas perancangan WithYou dengan menggunakan aplikasi dapat dilihat pada Gambar 5.3.

The screenshot shows the 'WithYou' application interface. At the top, there are three orange buttons: 'HASIL PENGGUNA', 'HASIL DEVELOPER', and 'HASIL DESAINER'. Below these is a section titled 'Urutan prioritas pembangunan aplikasi WithYou'. The features are listed in a numbered order, grouped by their value and cost characteristics:

- Fitur dengan Value Tinggi dan Cost Rendah:**
 1. Recommendation budget/price
 2. Search Vendor
- Fitur dengan Value Tinggi dan Cost Tinggi:**
 3. Calculate Budget
- Fitur dengan Value Rendah dan Cost Rendah:**
 4. Search Product
 5. Recommendation Product
- Fitur dengan Value Rendah dan Cost Tinggi:**
 6. Recommendation Vendor
 7. Shopping

Gambar 5.3 Hasil Prioritas WithYou

5.2.2 Pengujian Fungsionalitas

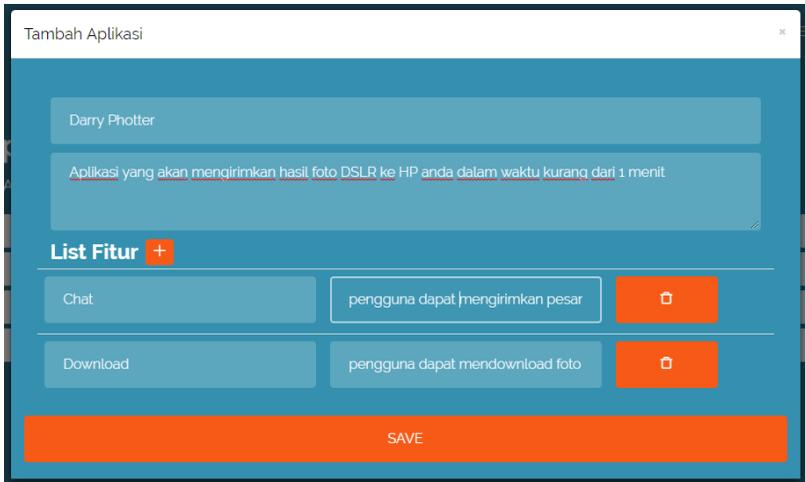
Pengujian fungsionalitas adalah pengujian kebutuhan fungsional yang dilakukan dengan metode *black box*. Pengujian ini menekankan pada pola *input* dan *output* aplikasi Priro sesuai dengan skenario kasus penggunaan yang ada pada Bab 3.

5.2.2.1 Pengujian Menambah Aplikasi

Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan menambahkan data aplikasi. Skenario pengujian dapat dilihat pada Tabel 5.11. Tampilan awal aplikasi dapat dilihat pada Gambar 5.4 dan hasil pengujian dari proses penambahan aplikasi dapat dilihat pada Gambar 5.5.

Tabel 5.11 Pengujian Menambah Aplikasi

ID	UJ-001
Kasus Penggunaan	Menambah Aplikasi
Tujuan Pengujian	Menguji fitur menambah aplikasi
Skenario	<i>Product Owner</i> melakukan penambahan aplikasi
Kondisi Awal	<i>Product Owner</i> telah melakukan <i>login</i>
Langkah Pengujian	<ol style="list-style-type: none">1. <i>Product Owner</i> menekan tombol +2. <i>Product Owner</i> mengisikan nama dan deskripsi aplikasi3. <i>Product Owner</i> menekan tombol <i>Save</i>
Hasil yang Diharapkan	Sistem menambahkan data aplikasi baru ke tabel aplikasi
Hasil yang Didapat	Sistem berhasil menambahkan data ke tabel aplikasi
Hasil Pengujian	Berhasil



Gambar 5.4 Pengujian Menambah Aplikasi



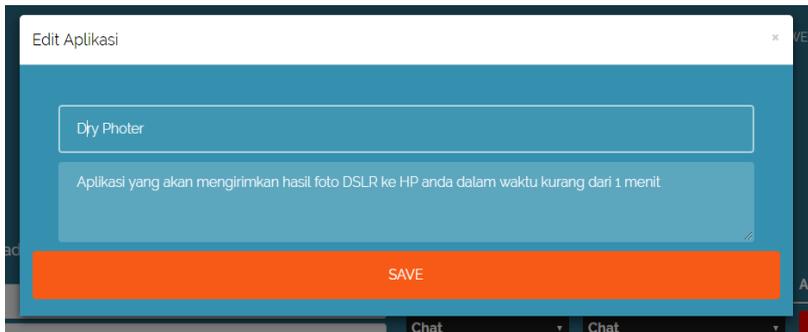
Gambar 5.5 Hasil Pengujian Menambah Aplikasi

5.2.2.2 Pengujian Mengubah Data Aplikasi

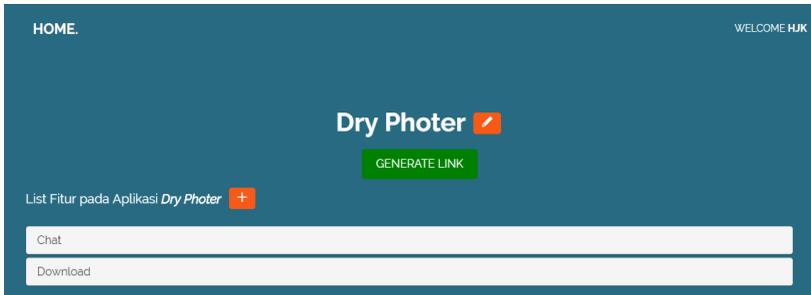
Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan mengubah data aplikasi. Skenario pengujian dapat dilihat pada Tabel 5.12. Tampilan awal aplikasi dapat dilihat pada Gambar 5.6 dan hasil pengujian dari proses mengubah data aplikasi dapat dilihat pada Gambar 5.7.

Tabel 5.12 Pengujian Mengubah Aplikasi

ID	UJ-002
Kasus Penggunaan	Mengubah Aplikasi
Tujuan Pengujian	Menguji fitur mengubah data aplikasi
Skenario	<i>Product Owner</i> melakukan perubahan data aplikasi
Kondisi Awal	<i>Product Owner</i> telah menekan tombol <i>edit</i> aplikasi
Langkah Pengujian	<ol style="list-style-type: none">1. <i>Product Owner</i> menekan tombol <i>edit</i> aplikasi2. <i>Product Owner</i> mengubah nama dan deskripsi aplikasi3. <i>Product Owner</i> menekan tombol <i>Save</i>
Hasil yang Diharapkan	Sistem mengubah data aplikasi yang sudah ada pada <i>database</i>
Hasil yang Didapat	Sistem berhasil mengubah data pada tabel aplikasi
Hasil Pengujian	Berhasil



Gambar 5.6 Pengujian Mengubah Aplikasi



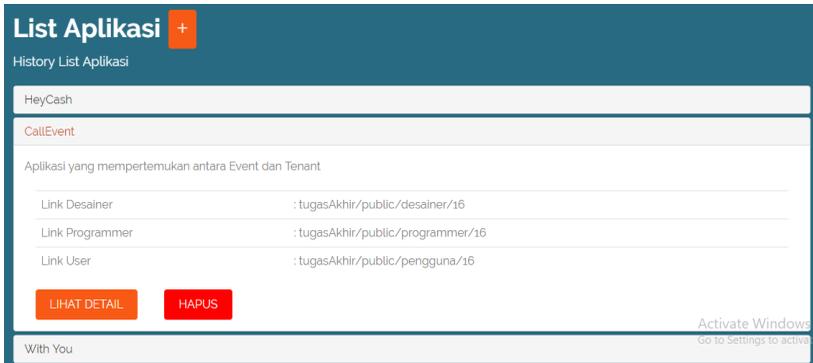
Gambar 5.7 Hasil Pengujian Mengubah Aplikasi

5.2.2.3 Pengujian Menghapus Aplikasi

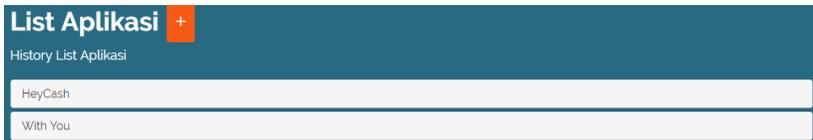
Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan menghapus data aplikasi. Skenario pengujian dapat dilihat pada Tabel 5.13. Tampilan awal aplikasi dapat dilihat pada Gambar 5.8 dan hasil pengujian dari proses menghapus data aplikasi dapat dilihat pada Gambar 5.9.

Tabel 5.13 Pengujian Menghapus Aplikasi

ID	UJ-003
Kasus Penggunaan	Menghapus Aplikasi
Tujuan Pengujian	Menguji fitur menghapus data aplikasi
Skenario	<i>Product Owner</i> melakukan penghapusan pada data aplikasi
Kondisi Awal	<i>Product Owner</i> telah menekan tombol hapus aplikasi
Langkah Pengujian	1. <i>Product Owner</i> menekan tombol hapus aplikasi
Hasil yang Diharapkan	Sistem menghapus data aplikasi yang sudah ada pada <i>database</i>
Hasil yang Didapat	Sistem berhasil menghapus data pada tabel aplikasi
Hasil Pengujian	Berhasil



Gambar 5.8 Pengujian Menghapus Aplikasi



Gambar 5.9 Hasil Pengujian Menghapus Aplikasi

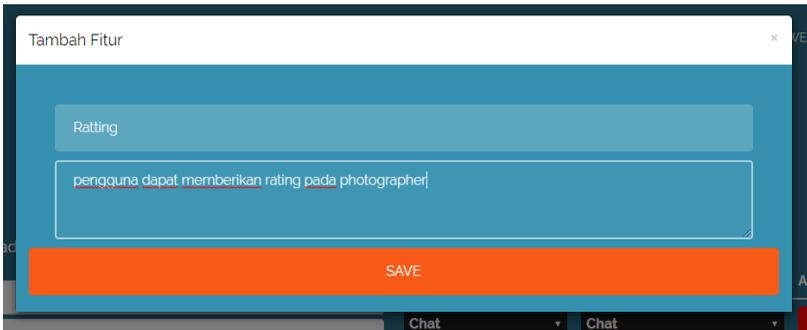
5.2.2.4 Pengujian Menambah Fitur

Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan menambahkan data fitur. Skenario pengujian dapat dilihat pada Tabel 5.14. Tampilan awal aplikasi dapat dilihat pada Gambar 5.10 dan hasil pengujian dari proses menambah data fitur dapat dilihat pada Gambar 5.11.

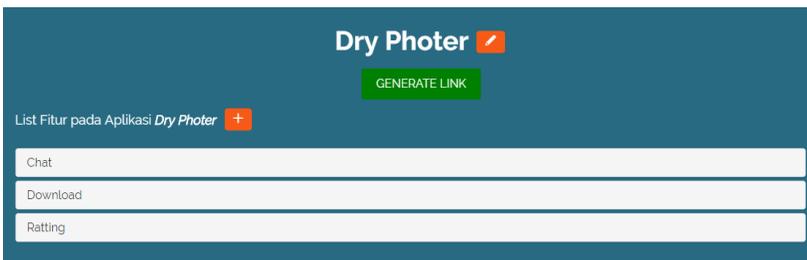
Tabel 5.14 Pengujian Menambah Fitur

ID	UJ-004
Kasus Penggunaan	Menambah Fitur
Tujuan Pengujian	Menguji fitur menambah fitur
Skenario	<i>Product Owner</i> melakukan penambahan fitur
Kondisi Awal	<i>Product Owner</i> berada pada halaman aplikasi
Langkah Pengujian	<ol style="list-style-type: none"> 1. <i>Product Owner</i> menekan tombol + 2. <i>Product Owner</i> mengisikan nama dan deskripsi fitur

	3. <i>Product Owner</i> menekan tombol <i>Save</i>
Hasil yang Diharapkan	Sistem menambahkan data fitur baru ke tabel fitur
Hasil yang Didapat	Sistem berhasil menambahkan data ke tabel fitur
Hasil Pengujian	Berhasil



Gambar 5.10 Pengujian Menambah Fitur



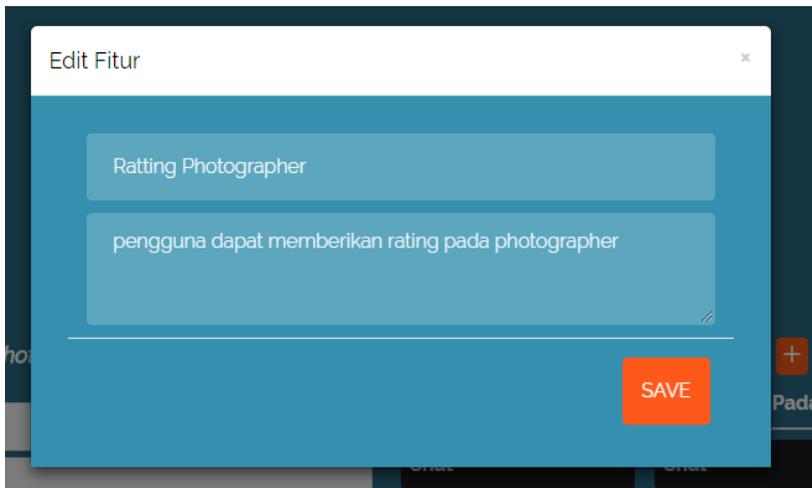
Gambar 5.11 Hasil Pengujian Menambah Fitur

5.2.2.5 Pengujian Mengubah Fitur

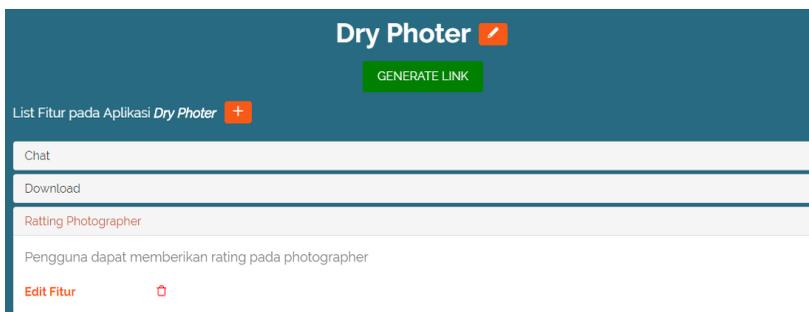
Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan mengubah data fitur. Skenario pengujian dapat dilihat pada Tabel 5.15. Tampilan awal aplikasi dapat dilihat pada Gambar 5.12 dan hasil pengujian dari proses mengubah data fitur dapat dilihat pada Gambar 5.13.

Tabel 5.15 Pengujian Mengubah Fitur

ID	UJ-005
Kasus Penggunaan	Mengubah Fitur
Tujuan Pengujian	Menguji fitur mengubah data fitur
Skenario	<i>Product Owner</i> melakukan perubahan data fitur
Kondisi Awal	<i>Product Owner</i> telah menekan tombol <i>Edit</i> Fitur
Langkah Pengujian	<ol style="list-style-type: none">1. <i>Product Owner</i> menekan tombol <i>Edit</i> Fitur2. <i>Product Owner</i> mengubah nama dan deskripsi fitur3. <i>Product Owner</i> menekan tombol <i>Save</i>
Hasil yang Diharapkan	Sistem mengubah data fitur yang sudah ada pada <i>database</i>
Hasil yang Didapat	Sistem berhasil mengubah data pada tabel fitur
Hasil Pengujian	Berhasil



Gambar 5.12 Pengujian Mengubah Fitur



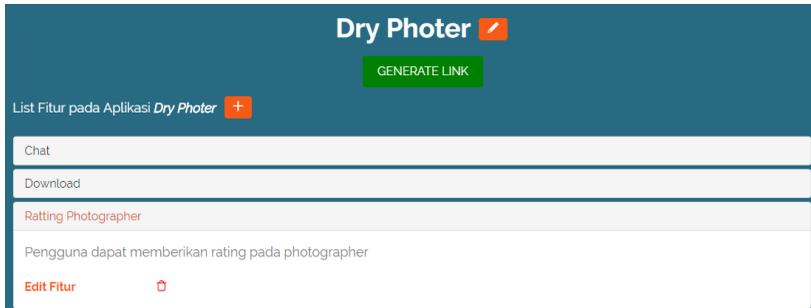
Gambar 5.13 Hasil Pengujian Mengubah Fitur

5.2.2.6 Pengujian Menghapus Fitur

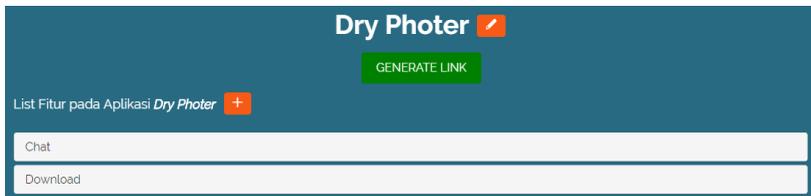
Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan menghapus data fitur. Skenario pengujian dapat dilihat pada Tabel 5.16. Tampilan awal aplikasi dapat dilihat pada Gambar 5.14 dan hasil pengujian dari proses menghapus data fitur dapat dilihat pada Gambar 5.15.

Tabel 5.16 Pengujian Menghapus Fitur

ID	UJ-006
Kasus Penggunaan	Menghapus Fitur
Tujuan Pengujian	Menguji fitur menghapus data fitur
Skenario	<i>Product Owner</i> melakukan penghapusan pada data fitur
Kondisi Awal	<i>Product Owner</i> telah menekan tombol berbentuk tempat sampah
Langkah Pengujian	1. <i>Product Owner</i> menekan tombol berbentuk tempat sampah
Hasil yang Diharapkan	Sistem menghapus data fitur yang sudah ada pada <i>database</i>
Hasil yang Didapat	Sistem berhasil menghapus data dari tabel fitur
Hasil Pengujian	Berhasil



Gambar 5.14 Pengujian Menghapus Fitur



Gambar 5.15 Hasil Pengujian Menghapus Fitur

5.2.2.7 Pengujian Melihat Jawaban Desainer

Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan melihat hasil jawaban desainer. Skenario pengujian dapat dilihat pada Tabel 5.17. Tampilan awal aplikasi dapat dilihat pada Gambar 5.20. dan hasil pengujian dari proses melihat jawaban desainer dapat dilihat pada Gambar 5.16.

Tabel 5.17 Pengujian Melihat Jawaban Desainer

ID	UJ-007
Kasus Penggunaan	Melihat Jawaban Desainer
Tujuan Pengujian	Menguji fitur melihat detail jawaban dari desainer
Skenario	<i>Product Owner</i> melihat detail jawaban dari desainer
Kondisi Awal	<i>Product Owner</i> telah menekan tombol HASIL DESAINER
Langkah Pengujian	1. <i>Product Owner</i> menekan tombol

	HASIL DESAINER 2. <i>Product Owner</i> memilih desainer yang ingin dilihat detail jawabannya
Hasil yang Diharapkan	Sistem menampilkan data hasil jawaban desainer
Hasil yang Didapat	Sistem berhasil menampilkan data hasil jawaban desainer
Hasil Pengujian	Berhasil

The screenshot shows a window titled "Detail Hasil Desainer" with a close button (x) in the top right corner. Below the title is a subtitle "- Akumulasi jawaban Desainer -". The main content is a table with two columns: "Nama Fitur" and "Persentase".

Nama Fitur	Persentase
Calculate Budget	31 %
Recommendation budget/price	275 %
Recommendation Product	75 %
Recommendation Vendor	10 %
Search Product	10 %
Search Vendor	10 %
Shopping	4 %

Gambar 5.16 Hasil Pengujian Melihat Jawaban Desainer

5.2.2.8 Pengujian Melihat Jawaban *Developer*

Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan melihat hasil jawaban *developer*. Skenario pengujian dapat dilihat pada Tabel 5.18. Tampilan awal aplikasi dapat dilihat pada Gambar 5.20 dan hasil pengujian dari proses melihat jawaban *developer* dapat dilihat pada Gambar 5.17.

Tabel 5.18 Pengujian Melihat Jawaban *Developer*

ID	UJ-008
Kasus Penggunaan	Melihat Jawaban <i>Developer</i>
Tujuan Pengujian	Menguji fitur melihat detail jawaban dari <i>developer</i>
Skenario	<i>Product Owner</i> melihat detail jawaban dari <i>developer</i>

Kondisi Awal	<i>Product Owner</i> telah menekan tombol HASIL DEVELOPER
Langkah Pengujian	1. <i>Product Owner</i> menekan tombol HASIL DEVELOPER
Hasil yang Diharapkan	Sistem menampilkan data hasil jawaban <i>developer</i>
Hasil yang Didapat	Sistem berhasil menampilkan data hasil jawaban <i>developer</i>
Hasil Pengujian	Berhasil

Detail Hasil Programmer

Nama : Marthin SP
Email : marthinpsrb@gmail.com

Nama Fitur	Waktu Pengerjaan	Biaya
Mencari Event	10 hari	Rp 2000000
Mencari Tenant	10 hari	Rp 2000000
Sistem Pembayaran	15 hari	Rp 15000000
Review	5 hari	Rp 1000000
Registrasi Booth	5 hari	Rp 1000000
Manage Tenant	7 hari	Rp 1000000

Gambar 5.17 Hasil Pengujian Melihat Jawaban *Developer*

5.2.2.9 Pengujian Melihat Jawaban Pengguna

Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan melihat hasil jawaban pengguna. Skenario pengujian dapat dilihat pada Tabel 5.19. Tampilan awal aplikasi dapat dilihat pada Gambar 5.20 dan hasil pengujian dari proses melihat jawaban desainer dapat dilihat pada Gambar 5.18.

Tabel 5.19 Pengujian Melihat Jawaban Pengguna

ID	UJ-009
Kasus Penggunaan	Melihat Jawaban Pengguna
Tujuan Pengujian	Menguji fitur melihat detail jawaban dari

	pengguna
Skenario	<i>Product Owner</i> melihat detail jawaban dari pengguna
Kondisi Awal	<i>Product Owner</i> telah menekan tombol HASIL PENGGUNA
Langkah Pengujian	<ol style="list-style-type: none"> 1. <i>Product Owner</i> menekan tombol HASIL PENGGUNA 2. <i>Product Owner</i> memilih pengguna yang ingin dilihat detail jawabannya
Hasil yang Diharapkan	Sistem menampilkan data hasil jawaban pengguna
Hasil yang Didapat	Sistem berhasil menampilkan data hasil jawaban pengguna
Hasil Pengujian	Berhasil

Detail Hasil Pengguna

- Akumulasi jawaban Pengguna -

Nama Fitur	Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
Manage Tenant	3	2	0	0
Mencari Event	4	1	0	0
Mencari Tenant	4	1	0	0
Registrasi Booth	4	1	0	0
Review	2	3	0	0
Sistem Pembayaran	3	2	0	0

- Hasil jawaban per Pengguna -

- Kezia Ayuningtyas (keziaay@gmail.com)
- Habibur Rahman (mhabiburr17@gmail.com)

Gambar 5.18 Hasil Pengujian Melihat Jawaban Pengguna

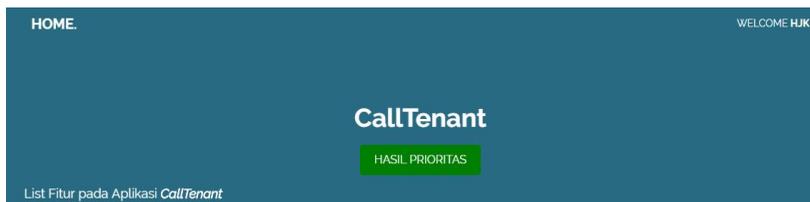
5.2.2.10 Pengujian Melihat Hasil Prioritas Aplikasi

Pada pengujian ini pengguna berperan sebagai *Product Owner* yang akan melihat hasil prioritas aplikasi. Skenario pengujian dapat dilihat pada Tabel 5.20. Tampilan awal aplikasi dapat dilihat

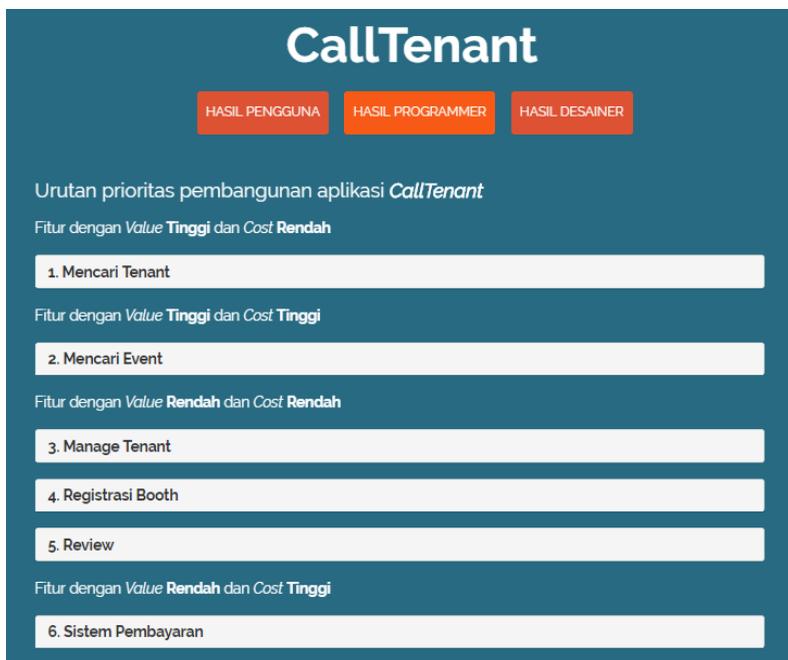
pada Gambar 5.20 dan hasil pengujian dari proses melihat hasil prioritas aplikasi dapat dilihat pada Gambar 5.19.

Tabel 5.20 Pengujian Melihat Hasil Prioritas Aplikasi

ID	UJ-010
Kasus Penggunaan	Melihat Hasil Prioritas Aplikasi
Tujuan Pengujian	Menguji fitur melihat hasil prioritas aplikasi
Skenario	<i>Product Owner</i> melihat hasil prioritas aplikasi
Kondisi Awal	<i>Product Owner</i> berada pada halaman aplikasi
Langkah Pengujian	<ol style="list-style-type: none"> 1. <i>Product Owner</i> menekan tombol HASIL PRIORITAS 2. <i>Product Owner</i> memilih fitur yang ingin dilihat detailnya
Hasil yang Diharapkan	Sistem menampilkan data hasil prioritas aplikasi
Hasil yang Didapat	Sistem berhasil menampilkan data hasil prioritas
Hasil Pengujian	Berhasil



Gambar 5.19 Tampilan Awal Melihat Hasil Prioritas Aplikasi



Gambar 5.20 Hasil Pengujian Melihat Hasil Prioritas Aplikasi

5.2.2.11 Pengujian Memberi Persentase Fitur

Pada pengujian ini pengguna berperan sebagai Desainer yang akan memberi persentase untuk setiap fitur pada aplikasi. Skenario pengujian dapat dilihat pada Tabel 5.21. Tampilan awal aplikasi dapat dilihat pada Gambar 5.21 dan hasil pengujian dari proses memberi persentase fitur dapat dilihat pada Gambar 5.16.

Tabel 5.21 Pengujian Memberi Persentase Fitur

ID	UJ-011
Kasus Penggunaan	Memberi Persentase Fitur
Tujuan Pengujian	Menguji fitur memberi persentase fitur
Skenario	Desainer memberikan persentase pada setiap fitur
Kondisi Awal	Desainer mengakses halaman untuk desainer

Langkah Pengujian	<ol style="list-style-type: none"> 1. Desainer mengakses halaman desainer 2. Desainer mengisikan data diri 3. Desainer menekan tombol Lanjutkan 4. Desainer memberikan persentase pada setiap fitur 5. Desainer menekan tombol <i>Save</i>
Hasil yang Diharapkan	Sistem menyimpan data persentase baru dari desainer
Hasil yang Didapat	Sistem berhasil menyimpan data baru persentase dari desainer
Hasil Pengujian	Berhasil

- CallEvent -

Aplikasi yang mempertemukan antara Event dan Tenant

Hi **Hapus** Anda akan memberikan persentasi kepentingan pada setiap fitur **CallEvent**

Total akumulasi persentase adalah **100!**

Nama Fitur	Deskripsi Fitur	Nilai
Mencari Event	pada fitur ini user mampu mencari dan melihat informasi mengenai event yang ada	ex:8
Mencari Tenant	EO dapat mencari tenant yang diinginkan untuk diajak bergabung pada event	ex:8
Sistem Pembayaran	Aplikasi mampu menangani sistem pembayaran dari tenant ke EO	ex:8

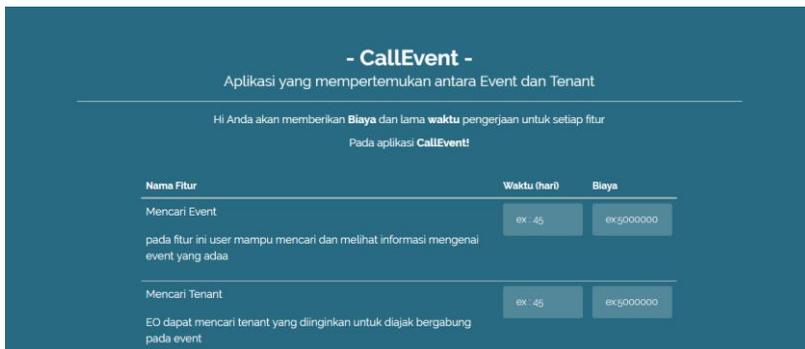
Gambar 5.21 Pengujian Memberi Persentase Fitur

5.2.2.12 Pengujian Memberi Waktu dan Biaya Fitur

Pada pengujian ini pengguna berperan sebagai *Developer* yang akan memberi waktu dan biaya pembangunan untuk setiap fitur pada aplikasi. Skenario pengujian dapat dilihat pada Tabel 5.22. Tampilan awal aplikasi dapat dilihat pada Gambar 5.22 dan hasil pengujian dari proses memberi waktu dan biaya fitur dapat dilihat pada Gambar 5.17.

Tabel 5.22 Pengujian Memberi Waktu dan Biaya Fitur

ID	UJ-012
Kasus Penggunaan	Memberi Waktu dan Biaya Fitur
Tujuan Pengujian	Menguji fitur memberi waktu dan biaya pada fitur
Skenario	<i>Developer</i> memberikan waktu dan biaya untuk setiap fitur
Kondisi Awal	<i>Developer</i> mengakses halaman untuk <i>developer</i>
Langkah Pengujian	<ol style="list-style-type: none">1. <i>Developer</i> mengakses halaman <i>developer</i>2. <i>Developer</i> mengisikan data diri3. <i>Developer</i> menekan tombol Lanjutkan4. <i>Developer</i> memberikan waktu dan biaya untuk setiap fitur5. <i>Developer</i> menekan tombol <i>Save</i>
Hasil yang Diharapkan	Sistem menyimpan data waktu dan biaya baru dari <i>developer</i>
Hasil yang Didapat	Sistem berhasil menyimpan data waktu dan biaya baru dari <i>developer</i>
Hasil Pengujian	Berhasil



Gambar 5.22 Pengujian Memberi Waktu dan Biaya Fitur

5.2.2.13 Pengujian Memberi Nilai Fitur

Pada pengujian ini pengguna berperan sebagai Pengguna yang akan memberi nilai untuk setiap fitur pada aplikasi. Skenario pengujian dapat dilihat pada Tabel 5.23. Tampilan awal aplikasi dapat dilihat pada Gambar 5.23 dan hasil pengujian dari proses memberi nilai fitur dapat dilihat pada Gambar 5.18.

Tabel 5.23 Pengujian Memberi Nilai Fitur

ID	UJ-013
Kasus Penggunaan	Memberi Nilai Fitur
Tujuan Pengujian	Menguji fitur memberi nilai pada fitur
Skenario	Pengguna memberikan nilai untuk setiap fitur
Kondisi Awal	Pengguna mengakses halaman untuk pengguna
Langkah Pengujian	<ol style="list-style-type: none">1. Pengguna mengakses halaman pengguna2. Pengguna mengisikan data diri3. Pengguna menekan tombol Lanjutkan4. Pengguna memberikan nilai untuk setiap fitur5. Pengguna menekan tombol <i>Save</i>
Hasil yang Diharapkan	Sistem menyimpan data nilai yang baru dari pengguna
Hasil yang Didapat	Sistem berhasil menyimpan data nilai yang baru dari pengguna
Hasil Pengujian	Berhasil

- CallEvent -	
Aplikasi yang mempertemukan antara Event dan Tenant	
Hi Anda akan memberikan Pendapat anda untuk setiap fitur	
Pada aplikasi CallEvent	
Nama Fitur	Pendapat anda tentang kebutuhan fitur tersebut untuk dibangun
Mencari Event pada fitur ini user mampu mencari dan melihat informasi mengenai event yang ada	<input checked="" type="radio"/> Sangat Dibutuhkan <input checked="" type="radio"/> Dibutuhkan <input type="radio"/> Mungkin Dibutuhkan <input type="radio"/> Tidak Dibutuhkan
Mencari Tenant EO dapat mencari tenant yang diinginkan untuk diajak bergabung pada event	<input checked="" type="radio"/> Sangat Dibutuhkan <input checked="" type="radio"/> Dibutuhkan <input type="radio"/> Mungkin Dibutuhkan <input type="radio"/> Tidak Dibutuhkan

Gambar 5.23 Pengujian Memberi Nilai Fitur

5.3 Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian perhitungan metode yang sudah dilakukan.

5.3.1 Evaluasi Pengujian Perancangan Suatu Perangkat Lunak dengan Bantuan Aplikasi

Hasil pengujian metode terhadap suatu perangkat lunak akan dijelaskan pada subbab ini. Perangkat lunak atau aplikasi yang sudah dilakukan uji coba dengan metode ini adalah aplikasi CallTenant dan WithYou. Berdasarkan hasil uji coba yang dijelaskan pada subbab 5.2.1 evaluasi pengujiannya dapat dilihat pada subbab ini.

5.3.1.1 Evaluasi Pengujian Perhitungan Metode

Hasil prioritas yang ditampilkan pada aplikasi Priro merupakan hasil perhitungan AHP menggunakan algoritma yang dilampirkan pada Bab 2. Untuk melihat apakah pengaplikasian perhitungan dalam aplikasi sudah berjalan dengan benar, dilakukan evaluasi dengan membandingkan hasil akhir yang ditampilkan aplikasi Priro dan aplikasi perhitungan AHP lainnya. Aplikasi AHP yang dijadikan pembanding aplikasi yang sudah dibangun ini tersedia pada: https://bpmmsg.com/academic/ahp_calc.php

Hasil akhir menunjukkan bahwa urutan prioritas hasil aplikasi Priro dan aplikasi perhitungan AHP adalah sama. Namun, didapati perbedaan persentase perhitungannya yaitu kurang lebih 0.2 untuk per fiturnya.

```
array:6 [▼  
  0 => {#258 ▼  
    +"nama_fitur": "Mencari Event"  
    +"valueahp*100": 20.850399136543  
  }  
  1 => {#255 ▼  
    +"nama_fitur": "Mencari Tenant"  
    +"valueahp*100": 43.272820115089  
  }  
  2 => {#249 ▼  
    +"nama_fitur": "Sistem Pembayaran"  
    +"valueahp*100": 5.7569768279791  
  }  
  3 => {#259 ▼  
    +"nama_fitur": "Review"  
    +"valueahp*100": 3.8944337517023  
  }  
  4 => {#254 ▼  
    +"nama_fitur": "Registrasi Booth"  
    +"valueahp*100": 13.112685084343  
  }  
  5 => {#252 ▼  
    +"nama_fitur": "Manage Tenant"  
    +"valueahp*100": 13.112685084343  
  }  
}
```

1
Gambar 5.24 Hasil AHP value pada aplikasi

based on your previous comparisons

Category	Priority	Rank
1 Mencari Event	20.9%	2
2 Mencari Tenant	43.9%	1
3 Sistem Pembayaran	5.6%	5
4 Review	3.8%	6
5 Registrasi Booth	12.9%	3
6 Manage Tenant	12.9%	3

Gambar 5.25 Hasil AHP pada aplikasi lain

5.3.1.2 Evaluasi Pengujian Perancangan CallTenant

Berdasarkan jawaban dari pengguna, *developer* dan desainer pada Tabel 5.3, Tabel 5.4, dan Tabel 5.5 Priro mendapatkan hasil akhir seperti pada Gambar 5.2. Hasil prioritas yang diberikan oleh aplikasi disandingkan dengan urutan pembangunan aplikasi yang sudah dirancang oleh desainer CallTenant. Pada Tabel 5.24 menjelaskan tentang hasil perhitungan aplikasi dan aplikasi AHP yang sudah ada terhadap fitur dari CallTenant. Tabel 5.25 merupakan hasil urutan dari perhitungan aplikasi Priro. Tabel 5.26 menunjukkan hasil bahwa empat dari enam fitur yang dirancang memiliki kesesuaian dengan perancangan yang dilakukan oleh desainer. Gambar 5.26 merupakan visualisasi perbandingan *cost* dan *value* dari hasil perhitungan aplikasi ke dalam bentuk graf.

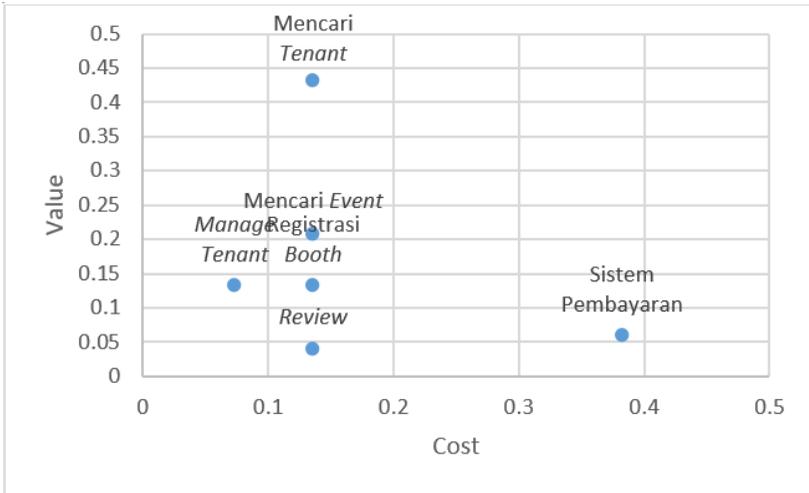
Perancangan yang diberikan oleh desainer merupakan rancangan yang dilakukan sebelum pembangunan aplikasi dilakukan. Namun, dalam implementasi pembangunan CallTenant urutan pengerjaan setiap fiturnya dapat dilihat pada Tabel 5.27. Di mana urutan pengerjaan fitur sesuai dengan hasil dari aplikasi dengan dua fitur yang dikerjakan secara bersamaan.

Tabel 5.24 Perbandingan Hasil Aplikasi Priro dan Aplikasi Lainnya

<i>Ranking Hasil Aplikasi</i>	<i>Persentase value Hasil Aplikasi</i>	<i>Nama Fitur</i>	<i>Persentase Hasil Aplikasi Lainnya</i>	<i>Ranking Hasil Aplikasi Lainnya</i>
1	43.27%	<i>Mencari Tenant</i>	43.9%	1
2	20.85%	<i>Mencari Event</i>	20.9%	2
3	13.11%	<i>Registrasi Booth</i>	12.9%	3
3	13.11%	<i>Manage Tenant</i>	12.9%	3
5	3.89%	<i>Review</i>	3.8%	5
4	5.76%	<i>Sistem Pembayaran</i>	5.6%	4

Tabel 5.25 Tabel Hasil Aplikasi Priro

<i>Ranking Hasil Aplikasi</i>	<i>value Hasil Aplikasi</i>	<i>cost Hasil Aplikasi</i>	<i>Nama Fitur</i>
1	0.433	0.136	<i>Mencari Tenant</i>
2	0.209	0.136	<i>Mencari Event</i>
3	0.131	0.136	<i>Registrasi Booth</i>
4	0.131	0.073	<i>Manage Tenant</i>
5	0.039	0.136	<i>Review</i>
6	0.058	0.382	<i>Sistem Pembayaran</i>



Gambar 5.26 Graf Hasil Aplikasi CallTenant

Tabel 5.26 Hasil Jawaban Desainer

No Urut Aplikasi	Urutan Hasil Akhir Aplikasi	Urutan Perancangan oleh Desainer	Hasil
1.	Mencari <i>Tenant</i>	Mencari <i>Tenant</i>	Sesuai
2.	Mencari <i>Event</i>	Mencari <i>Event</i>	Sesuai
3.	Registrasi <i>Booth</i>	Registrasi <i>Booth</i>	Sesuai
4.	<i>Manage Tenant</i>	<i>Manage Tenant</i>	Sesuai
5.	<i>Review</i>	Sistem Pembayaran	Tidak Sesuai
6.	Sistem Pembayaran	<i>Review</i>	Tidak Sesuai

Tabel 5.27 Urutan Pembangunan Fitur Berdasarkan Jawaban Desainer Aplikasi CallTenant

No Urut	Urutan Pengerjaan Fitur	Keterangan
1.	Mencari <i>Tenant</i>	Dikerjakan bersamaan
1.	Mencari <i>Event</i>	
2.	<i>Manage Tenant</i>	Dikerjakan bersamaan
2.	Registrasi <i>Booth</i>	
3.	<i>Review</i>	
4.	Sistem Pembayaran	

Dari hasil uji coba dengan perangkat lunak CallTenant, dapat dilihat bahwa urutan awal perancangan yang sudah dirancang oleh desainer memang sesuai dengan perhitungan *value* dari aplikasi *online* perhitungan AHP. Namun, dalam pembangunan aplikasinya sendiri, urutannya mengikuti urutan yang dihasilkan oleh aplikasi yang sudah dibangun.

Setelah melakukan konfirmasi kepada tim *startup* yang membangun CallTenant ditemukan adanya alasan mengapa perancangan urutan fitur di awal berbeda dengan pengimplementasiannya. Dijelaskan bahwa pembangunan fitur sistem pembayaran dilakukan paling terakhir karena fitur tersebut tidak termasuk dalam kategori fitur utama atau yang paling penting dan sistem pembayaran merupakan fitur dengan biaya dan waktu pengerjaan yang paling besar [12].

Aplikasi prioritas Priro tidak hanya mempertimbangkan *value* saja, tetapi juga turut memperhitungkan *cost*. Oleh sebab itu, urutan prioritas fitur CallTenant dari Priro sesuai dengan urutan kenyataan pembangunannya.

5.3.1.3 Evaluasi Pengujian Perancangan WithYou

Berdasarkan jawaban dari pengguna, *developer* dan desainer pada Tabel 5.8, Tabel 5.9 dan Tabel 5.10 Priro mendapatkan hasil akhir seperti pada Gambar 5.3. Hasil prioritas yang diberikan oleh

aplikasi dibandingkan dengan urutan pembangunan aplikasi yang sudah dirancang oleh desainer WithYou. Pada Tabel 5.28 menjelaskan tentang hasil perhitungan aplikasi dan aplikasi AHP yang sudah ada terhadap fitur dari WithYou. Tabel 5.29 merupakan hasil urutan dari perhitungan aplikasi Priro. Tabel 5.30 menunjukkan hasil bahwa fitur yang dirancang memiliki kesesuaian dengan perancangan yang dilakukan oleh desainer. Gambar 5.27 merupakan visualisasi perbandingan *cost* dan *value* dari hasil perhitungan aplikasi ke dalam bentuk graf.

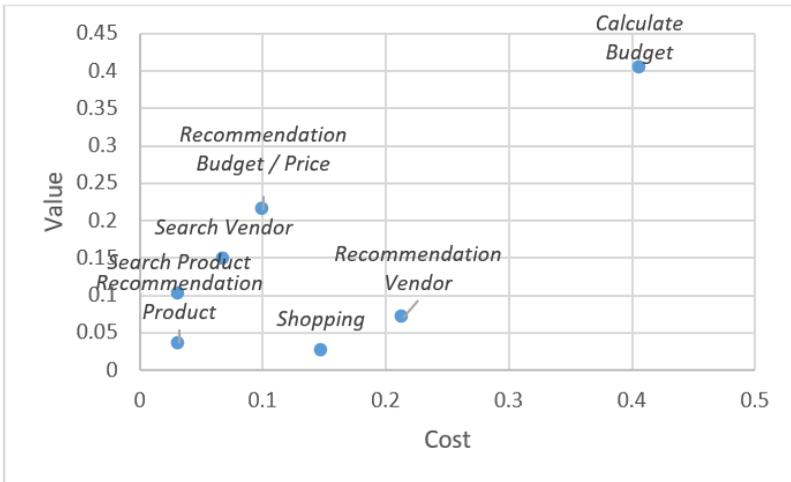
Perancangan yang diberikan oleh desainer merupakan rancangan yang dilakukan sebelum pembangunan aplikasi dilakukan. Namun, dalam implementasi pembangunan WithYou urutan pengerjaan setiap fiturnya dapat dilihat pada Tabel 5.31. Di mana terlihat bahwa ada beberapa fitur yang dibangun secara bersamaan dan ada fitur yang masih belum dibangun atau akan dibangun di masa depan.

Tabel 5.28 Perbandingan Hasil Aplikasi dan Aplikasi Lainnya

<i>Ranking</i> Hasil Aplikasi	Persentase <i>value</i> Hasil Aplikasi	Nama Fitur	Persentase Hasil Aplikasi Lainnya	<i>Ranking</i> Hasil Aplikasi Lainnya
1	40.53%	<i>Calculate Budget</i>	41.4%	1
3	14.79%	<i>Search Vendor</i>	14.6%	3
5	7.03%	<i>Recommendation Vendor</i>	6.7%	5
4	10.16%	<i>Search Product</i>	9.8%	4
6	3.55%	<i>Recommendation Product</i>	3.4%	6
2	21.40%	<i>Recommendation Budget / Price</i>	21.6%	2
7	2.53%	<i>Shopping</i>	2.4%	7

Tabel 5.29 Tabel Hasil Aplikasi Priro

<i>Ranking Hasil Aplikasi</i>	<i>value Hasil Aplikasi</i>	<i>cost Hasil Aplikasi</i>	<i>Nama Fitur</i>
1	0.14794	0.068336	<i>Recommendation Budget / Price</i>
2	0.070304	0.213945	<i>Search Vendor</i>
3	0.405338	0.406045	<i>Calculate Budget</i>
4	0.101582	0.03202	<i>Search Product</i>
5	0.035526	0.03202	<i>Recommendation Product</i>
6	0.214018	0.100318	<i>Recommendation Vendor</i>
7	0.025292	0.147316	<i>Shopping</i>



Gambar 5.27 Graf Hasil Aplikasi WithYou

Tabel 5.30 Hasil Jawaban Desainer

No Urut Aplikasi	Urutan Hasil Aplikasi	Urutan Perancangan oleh Desainer	Hasil
1.	<i>Recommendation Budget / Price</i>	<i>Recommendation Budget / Price</i>	Sesuai
2.	<i>Search Vendor</i>	<i>Search Vendor</i>	Sesuai
3.	<i>Calculate Budget</i>	<i>Calculate Budget</i>	Sesuai
4.	<i>Search Product</i>	<i>Search Product</i>	Sesuai
5.	<i>Recommendation Product</i>	<i>Recommendation Product</i>	Sesuai
6.	<i>Recommendation Vendor</i>	<i>Recommendation Vendor</i>	Sesuai
7.	<i>Shopping</i>	<i>Shopping</i>	Sesuai

Tabel 5.31 Urutan Pembangunan Fitur Berdasarkan Jawaban Desainer Aplikasi WithYou

No Urut	Urutan Pengerjaan Fitur	Keterangan
1.	<i>Search Vendor</i>	Dikerjakan
2.	<i>Recommendation Vendor</i>	Bersamaan
3.	<i>Search Product</i>	Dikerjakan
4.	<i>Recommendation Product</i>	Bersamaan
5.	<i>Calculate Budget</i>	
6.	<i>Shopping</i>	
7.	<i>Recommendation Budget / Price</i>	<i>Coming Soon</i>

Dari hasil uji coba dengan perangkat lunak WithYou, dapat dilihat bahwa urutan awal perancangan yang sudah dirancang oleh desainer sesuai dengan hasil yang didapatkan dari aplikasi prioritas yang sudah dibangun. Namun, dalam pengaplikasiannya saat pembangunan fitur WithYou urutan pengerjaannya berbeda.

Setelah melakukan konfirmasi kepada tim *startup* yang membangun WithYou ditemukan alasan mengapa urutan pembangunan aplikasi berbeda dengan rancangan yang dirancang dari awal. Alasan yang diberikan adalah adanya kendala sumber daya manusia yang akan membangun aplikasi ini atau *developer*. Kurangnya *developer* untuk membangun perangkat lunak ini menyebabkan pembangunan aplikasi dilakukan dengan kemampuan *developer* yang ada. Selain itu, tuntutan waktu pembangunan aplikasi menjadi salah satu faktor yang mempengaruhi urutan pengerjaan setiap fitur pada akhirnya [13].

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran yang dapat diberikan setelah melakukan uji coba pada bab sebelumnya. Saran pada bab ini juga ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.

6.1 Kesimpulan

Dari proses pengerjaan Tugas Akhir ini, setelah melakukan proses perancangan, implementasi, dan pengujian perangkat lunak, dapat diambil kesimpulan sebagai berikut:

1. Aplikasi ini mampu memberikan rekomendasi berupa urutan daftar fitur yang akan dibangun.
2. Aplikasi Priro mampu mengolah data dari tiga elemen berbeda yaitu desainer, *developer*, dan pengguna untuk menghitung prioritas.
3. Setelah dilakukan perhitungan dengan aplikasi Priro, adanya ketidaksesuaian terjadi pada fitur-fitur tambahan atau yang kurang penting.
4. Aplikasi menampilkan hasil yang berbeda dari perspektif AHP pada umumnya karena aplikasi Priro memiliki faktor tambahan yaitu biaya dan waktu pengerjaan setiap fitur.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi, dan pengujian yang telah dilakukan.

1. Tingkat ketergantungan antar fitur ikut menjadi pertimbangan dalam penentuan prioritas.
2. Penambahan fitur pengelolaan data hasil jawaban dari pengguna, desainer, dan *developer*.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] Babar, M. I., Ghazali, M., Jawawi, D.N.A., Shamsuddin, S. M., Ibrahim, N., PHandler: An expert system for a scalable software requirements prioritization process, Knowledge-Based System, Volume 84, p179-202, 2013.
- [2] “4 Requirements Prioritization.” [Daring]. Tersedia pada: <https://www.divaportal.org/smash/get/diva2:836447/FULLTEXT01.pdf> . [Diakses pada 11 Agustus 2017].
- [3] Knapp, Jake. Kowitz, Braden. 2016. Sprint: How to Solve Big Problems and Test New Ideas in Just Five Day.
- [4] “Analytic Hierarchy Process.” [Daring]. Tersedia pada: <http://www.pwktech.info/systemmodeling/analytic-hierarchy-process-ahp/> . [Diakses pada 26 Mei 2017].
- [5] “Comparison Analysis of Requirement Prioritization Methods Between Case Based Ranking and Cumulative Voting.” [Daring]. Tersedia pada: <https://yudiagusta.files.wordpress.com/2008/09/022-027-knsi2011-004comparison-analysis-of-requirement-prioritization-methods-between-case-basedranking-and-cumulative-voting.pdf> . [Diakses pada 11 Agustus 2017].
- [6] Mu, E., Rojas, M.P., 2017. Practical Decision Making An Introduction to the Analytic Hierarchy Process (AHP) Using Super Decisions V2. Springer International Publishing .
- [7] “MoSCoW Method: the Most Successful Prioritization Technique For Any Project.” [Daring]. Tersedia pada: <https://blog.ganttpro.com/en/prioritization-techniques-and-methods-for-projects-with-advantages-of-moscow-model/>. [Diakses pada 30 Desember 2017] .
- [8] “Introduction - Laravel.” [Daring]. Tersedia pada: <https://laravel.com/docs/4.2/introduction> . [Diakses pada 19 September 2017].
- [9] Hidayat, A., Surarso, B., 2012. Penerapan Arsitektur *Model View Controller* (MVC) dalam Raancang Bangun Sistem

Kuis *Online* Adaptif. Seminar Nasional Teknologi Informasi dan Komunikasi 2012.

- [10] Christofer, Sun Win wawancara. 2017. “Penjelasan mengenai *startup* CallTenant”. Indonesia.
- [11] Oo, Thanda wawancara. 2017. “Penjelasan mengenai *startup* WithYou”. Korea Selatan.
- [12] Pasaribu, Marthin wawancara. 2017. “Teknis pembangunan aplikasi CallTenant”. Indonesia.
- [13] Kamilah, Hilma wawancara. 2017. “Teknis pembangunan aplikasi WithYou”. Indonesia.

LAMPIRAN

Bagian ini merupakan pelengkap dari pengerjaan Tugas Akhir ini. Pada Tabel A.1 dijelaskan jawaban hasil kuesioner yang ditanyakan kepada pelaku *startup* mengenai perbandingan persentase tingkat kepentingan antara desainer dan pengguna aplikasi.

Tabel A.1. Jawaban Hasil Kuesioner

Nama	Nama <i>Startup</i>	Desainer	Pengguna
Marthin Satrya	Medan Digital Center	35	65
Adi Wibowo P.	WZTruk	70	30
Azkaa Khoiruddin	GLUE	60	40
Nida Amalia	Edubraille	20	80
Kevin	Medify	40	60
Anam	Lixense	40	60
Annisa Bunga Nafara	Smartmom	30	70
Muhammad Sarwani	Smartmom	40	60
Tegar Djati Prayogo	Lixense	10	90
Muhammad Fahlul	MYPITCH	30	70

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Anne Annisa Aulia, lahir di Jakarta pada tanggal 16 Desember 1995. Biasa dipanggil Anne oleh teman – teman. Penulis merupakan anak pertama dari dua bersaudara. Penulis menempuh pendidikan formal yang dimulai dari TK Mini Pak Kasur, SD Islam Al-Azhar 9 Kemang Pratama, SMP Islam Al-Azhar 8 Kemang Pratama, SMA Negri 81 Jakarta. Dilanjutkan dengan jenjang S1 di Institut Teknologi Sepuluh Nopember dengan mengambil Jurusan Teknik Informatika. Rumpun mata kuliah yang dipilih oleh penulis adalah Manajemen Informasi dan penulis turut aktif menjadi administrator Laboratorium Manajemen Informasi di Jurusan Teknik Informatika ITS. Selama masa kuliah, penulis tergabung dalam beberapa kepanitiaan baik tingkat jurusan ataupun institut, penulis juga ikut serta dalam beberapa komunitas dan berkesempatan menjadi penerima beasiswa dari Bank Indonesia. Penulis sempat menjadi asisten dosen pada beberapa matakuliah. Penulis berkesempatan mengikuti program ICT Global *Startup* 2017 yang diadakan di Korea Selatan untuk mempelajari *Startup*. Penulis memiliki ketertarikan pada bidang *Startup* dan bisnis IT. Komunikasi dengan penulis dapat dilakukan melalui *email* **anneannisa@hotmail.com**